Software Engineering Stage 6 (Year 12) – teacher support resource

**Programming for the web**

# Teacher support resource

**Teacher note:** this resource has been designed to facilitate the ready conversion into a student booklet by removing the answers within the response windows. Teacher notes can be deleted before distributing to students. This booklet should be submitted as the documentation component of the assessment task in part or in whole.

Student name:

Class:

Teacher:

Contents

[Teacher support resource 2](#_Toc175911953)

[Unit overview 5](#_Toc175911954)

[Task description 8](#_Toc175911955)

[Project description 9](#_Toc175911956)

[Progressive web apps (PWAs) 9](#_Toc175911957)

[Submission details 10](#_Toc175911958)

[Component A – project documentation 10](#_Toc175911959)

[Component B – software solution 11](#_Toc175911960)

[Component C – presentation 11](#_Toc175911961)

[Steps to success 12](#_Toc175911962)

[Glossary 14](#_Toc175911963)

[NESA glossary keywords 19](#_Toc175911964)

[Programming for the web pre-test 20](#_Toc175911965)

[Data transmission using the web 24](#_Toc175911966)

[Designing web applications 46](#_Toc175911967)

[Appendix 1 – progressive web application 77](#_Toc175911968)

[Introduction to PWA 77](#_Toc175911969)

[Set up your development environment 79](#_Toc175911970)

[Set up your development environment 80](#_Toc175911971)

[Establish a working directory 82](#_Toc175911972)

[Initialise Node.js application 82](#_Toc175911973)

[Set up files and folders for your project 83](#_Toc175911974)

[Build and test query your database 86](#_Toc175911975)

[Set up your Express webserver 88](#_Toc175911976)

[Render your front end 90](#_Toc175911977)

[Running an Express server via Node.js from VSCode 99](#_Toc175911978)

[Appendix 2 – web server guide 101](#_Toc175911979)

[Web server guide 102](#_Toc175911980)

[References 104](#_Toc175911981)

# Unit overview

This unit guides students through the Programming for the web focus area. The content for this focus area should be delivered as a project integrated with the theory required to develop the project. The theory activities go beyond those required to achieve the outcomes related to the progressive web app project (PWA), though could be used to inform a major project for this course. Teachers are encouraged to issue the PWA assessment task to students to work on alongside other lessons that deliver course content.

This **Teacher Support Resource (TSR)** provides activities that align with the content of the syllabus and ensures coverage. It is expected that after the initial delivery of this unit, in this linear and chronological approach, teachers will identify opportunities to provide theory and practice within single lessons. The Programming for the web is developed in the classroom under the supervision of the teacher using explicit teaching methods outlined in this TSR.

**Weeks 1 to 2**

* Students complete a pre-test on fundamental concepts within Programming for the web. This helps teachers determine the background knowledge and understanding of their class and which of their students may have completed the developing apps and web software focus area from the Computing Technology course. Teachers may allocate groups according to expertise based on the results of this pre-test.
* Students identify a project of personal interest or client for their PWA (suggested topics to catalogue: music, movies, videos, songs, websites, actors, games, restaurants, food, recipes, art, pop culture, quotes, canteen items, clothes and so on).
* Students complete activities that explore the applications of web programming. They investigate and practise how data is transferred on the internet. They describe the function of web protocols and their ports.
* For their PWA project, students set up their development environment,install the required VSCode extensions, install Node.js and install SQLite. They create a text file listing the core folders for the task. Students write and execute BASH scripts to routinely generate all files and folders.
* Students record the files being generated routinely in a video using OBS or similar.

**Weeks 3 to 4**

* Students explain the processes for securing the web. They investigate the effect of big data on web architecture. Students investigate and explain the role of the World Wide Web Consortium (W3C) in the development of applications for the web They model elements that form a web development system. Students explore and explain the influence of a web browser on web development, including the use of developer (dev) tools. They investigate cascading style sheets (CSS) and its impact on the design of a web application.
* For their project, students code the look of the front end. They create a wireframe or storyboard for the pages that will be present in the system. They create HTML and CSS files.

**Weeks 5 to 6**

* Students investigate the reasons for version control and apply it when developing web application.
* They explore the types and significance of code libraries for front-end web development.
* Students explain the use and development of open-source software in relation to web development.
* They investigate methods to support and manage the load times of webpages/applications.
* Students research, experiment with and evaluate the prevalence and use of web content management systems (CMS).
* They assess the contribution of back-end web development to the success of a web application.
* Students observe and describe the back-end process used to manage a web request.
* They continue to develop their web application using an appropriate scripting language with shell scripts to make files and directories and searching for text in a text file.
* Students plan and design the tables required within their database. For each table they create fields considering field names and data types. Students document SQL queries made in their SQL file by commenting them out progressively, so they are all recorded.
* Students apply a web-based database and construct script that executes SQL.
* For their PWA project, students build and test query their database. They code initial functionality by inserting data into their database and querying it to ensure data is present.

**Weeks 7 to 9**

* Students compare Object-Relational Mapping (ORM) to SQL.
* They describe how collaborative work practices between front-end and back-end developers improve the development of a web solution.
* For their PWA project, students set up their Express webserver.
* They continue to design, develop and implement their PWA by rendering the front end and coding remaining functionality. They build out their catalogue and query options.
* Students add icon sets and configure services.

**Week 10**

* Students test their PWA using W3C. They prepare and complete 'User accessibility testing’ document with a test scenario.
* Students present their PWA solution using presentation software to the class (and where relevant client). They submit their project documentation and code.

# Task description

**Type of task**: develop a software solution, project documentation and presentation for the Programming for web focus area.

**Outcomes being assessed**

A student:

* applies structural elements to develop programming code **SE-12-02**
* analyses how current hardware, software and emerging technologies influence the development of software engineering solutions **SE-12-03**
* justifies the selection and use of tools and resources to design, develop, manage and evaluate software **SE-12-06**
* designs, develops and implements safe and secure programming solutions **SE-12-07**
* tests and evaluates language structures to refine code **SE-12-08**

[Software Engineering 11–12 Syllabus](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/overview) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2022.

**Weighting**: 20%

Students plan, design and produce a PWA to catalogue a personal interest or address the needs of a client. The application should allow users to query the catalogue in a variety of ways.Suggested topics to catalogue include: music, movies, videos, songs, websites, actors, games, restaurants, food, recipes, art, pop culture, quotes, canteen items, sports, clothes and so on.

# Project description

**Teacher note:** students are encouraged to work with a critical friend to provide feedback and advice on user experience (UX) and user interface (UI) features and the functionality of the program. This task provides foundational knowledge and skills to inform students’ Software Engineering project. This personal interest project could be adapted and modified for a client. Teachers will need to discriminate between the submission of this assessment task and the final Software Engineering project.

## Progressive web apps (PWAs)

PWAs are a cross between a traditional website and a mobile application. They are built using web technologies like hyperlink markup language (HTML), cascading style sheets (CSS), JavaScript and web application programming interfaces (APIs), offering a user experience like a mobile app.

PWAs are popular because they are installable to all platforms using the same code base, they operate while offline and in the background. They can integrate with other services such as the device, other installed apps and access databases. Twitter and Hulu are the most well-known platforms utilising PWA standards

Students develop an understanding of the software engineer’s role in developing PWAs.

This includes engineering the manifest.json file to manage caching and installation behaviours, along with designing the PWA front-end user interface for the user to interact with and designing the back-end server to store data in a Structured Query Language (SQL) database.

Students plan, design and produce a progressive web app to catalogue a personal interest.

# Submission details

Students are to submit 3 components (A, B and C).

## Component A – project documentation

1. Process diary/logbook
2. Storyboard
3. Directory structure
4. Gantt chart

See [course specifications](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/teaching-and-learning) for standardised versions of project documentation. These need to be adhered to throughout the course and may form part of examinable content.

## Component B – software solution

1. UI
2. design and layout
3. evidence of accessibility and inclusivity considerations
4. evidence of usability considerations
5. Programming
6. HTML
7. CSS
8. JavaScript
9. SQL
10. Database
11. table creation (data types, field lengths, ordering)
12. queries
13. Testing
14. validation
15. user accessibility testing.

## Component C – presentation

1. Present their Software Engineering project to peers.
2. Respond to questions and answers (Q&A).

## Steps to success

Teachers should print [Appendix 1 – progressive web application](#_Appendix_1:_Progressive) of this teacher support resource and distribute to students to guide them in the completion of each of the steps to success.

Table 1 – assessment preparation schedule

|  |  |
| --- | --- |
| Steps | What I need to do |
| Set up your development environment | 1. Install the required VSCode extensions. 2. Install Node.js. 3. Install SQLite3. 4. Create a text file listing the core folders for the task. 5. Create a text file listing the core files for the task. 6. Write and execute BASH scripts to routinely generate all your files and folders.   **Note:** you should record the files being generated routinely in a video using OBS or similar. |
| Set up the database | Plan and design the tables required within the database. For each table, create fields, considering meaningful field names, and appropriate data types.  **Note:** all SQL queries made in your SQL file should be documented by commenting them out progressively, so they are all recorded. |
| Construct the front-end JSON file | Write a JavaScript algorithm that converts an SQL query into a JSON file and puts it to the front-end. |
| Code the look of the front end | 1. Create a wireframe or storyboard for the pages that will be present in the system. 2. Create HTML and CSS files. |
| Code initial functionality | 1. Insert data to your database. 2. Query your database to confirm the data is present. |
| Plan query options | Create an algorithm/IPO table outlining the different queries that will make accessing your catalogue more user-friendly. |
| Code remaining functionality | Using an agile development approach, build out the catalogue container and query options. |
| Implement progressive web app architecture | 1. Add icon sets 2. Configure services |
| Test | 1. Prepare and complete 'User accessibility testing’ document with test scenarios. 2. The following tools can be used to support testing: 3. [W3C](https://www.w3.org/WAI/ER/tools/) 4. [Google Lighthouse](https://developer.chrome.com/docs/lighthouse/overview/). |

# Glossary

Many of the following words will gather more meaning to you as you work through this booklet.

When you see one of the words from the table below, add its definition to the glossary in case you need to refer to them later.

Table 2 – glossary

|  |  |
| --- | --- |
| Word | Definition |
| **agile** | An iterative and flexible approach to software development, focusing on collaboration, customer feedback and small, rapid releases. |
| **algorithm** | A step-by-step procedure required to solve a problem. Algorithms may be presented in many ways, for example written instructions, flow charts or using a computer programming language. |
| **app** | A software program designed for a specific purpose to run on mobile devices or on a personal computer. An abbreviation of the word ‘application’. |
| **application programming interface (API)** | An interface that allows an application or website to plug into another program or website. |
| **boundary** | A boundary refers to the interface or demarcation line that separates one system or component from another. |
| **code** | Refers to the written instructions or statements in a programming language that define the logic, behaviour and functionality of a software application or system. Code is the fundamental building block of software development. |
| **code optimisation** | The process of improving the efficiency, performance and resource utilisation of software code without changing its external functionality or behaviour |
| **computational thinking** | A process in which a problem is analysed and solved so that a human, machine or computer can effectively implement the solution. It involves using strategies to organise data logically, break down problems into parts, interpret patterns and design and implement algorithms to solve problems. |
| **client** | A person or organisation using the services of a software engineer |
| **data flow diagrams** | Diagrams illustrating the flow of data within a system, showing how it is processed or transformed. |
| **data structures** | A data structure is a way of organising and storing data in a computer so that it can be accessed and used efficiently. |
| **data types** | Data types used in computing are expressed as either: string/text, character, integer, floating point or real, date and time, Boolean. |
| **decision trees** | Diagrams that represent decisions and their possible consequences, often used in decision-making processes. |
| **end-user** | The person or group who will ultimately use the system or product. |
| **Express** | A minimalist and flexible Node.js web application framework that simplifies the process of creating server-side logic, handling HTTP requests and defining routes |
| **Gantt charts** | Visual tools used for project management to represent the timing of tasks or activities. |
| **iterative approach** | A method of development where the project is divided into smaller parts and developed incrementally, with each iteration building on the previous one. |
| **object-oriented programming (OOP)** | A paradigm based on the concept of ‘objects’ that can contain data and code in the form of procedures. |
| **object-relational mapping(ORM)** | A technique used in object-oriented programming to query and manipulate data from a database. |
| **online collaboration** | Working together on projects or tasks using internet-based tools and platforms. |
| **outsourcing** | Hiring external individuals or companies to perform tasks or services instead of internal employees. |
| **Node.js:** | A JavaScript runtime environment that allows developers to run JavaScript code outside the browser. It provides a server-side platform for building scalable and efficient web applications. |
| **procedural programming** | A method of programming where the program is divided into functions. A program consists of data and procedures (modules) that operate on the data. Data and procedures are treated as separate entities. |
| **prototyping** | Building partial versions of a system to test ideas and gather feedback before full implementation. |
| **pseudocode** | A form of algorithm description that uses English-like statements with defined rules of structure and keywords. |
| **SQLite** | A lightweight, self-contained, serverless and open-source relational database management system (RDBMS) that is widely used in various software applications and embedded systems |
| **system flowcharts** | Visual representations of the flow of data or processes within a system. |
| **verification and validation** | Processes to ensure that the system meets specified requirements and functions correctly. |
| **VSCode:** | Visual Studio Code (VSCode) is a popular, lightweight and highly extensible code editor developed by Microsoft. It provides a rich set of features for coding, debugging, version control and integrating with various tools and frameworks. |
| **WAgile** | WAgile is a hybrid of waterfall and agile project management. |
| **waterfall (structured)** | A sequential development approach where progress flows steadily downward through predefined phases. |
| **working remotely** | Performing work from a location other than the traditional office setting, often enabled by technology and the internet. |

**Teacher note:** for students with an EALD background. The glossary can be provided complete so that they have additional time to understand the key terms with bilingual dictionaries. The glossary can be provided to students in their preferred communication mode.

## NESA glossary keywords

NESA keywords can be used in the syllabus and in the Higher School Certificate examination. Familiarisation with these keywords can assist in understanding how to write and respond to questions.

|  |  |
| --- | --- |
| Key term | Definition |
| Apply | Use, utilise, employ in a particular situation |
| Describe | Provide characteristics and features |
| Explain | Relate cause and effect; make the relationships between things evident; provide why and/or how |
| Investigate | Plan, inquire into and draw conclusions about |

[NESA: A Glossary of Key Words](https://educationstandards.nsw.edu.au/wps/portal/nesa/11-12/hsc/hsc-student-guide/glossary-keywords)

**Teacher note:** develop, explore, select and verify are used in this topic and are not listed.

# Programming for the web pre-test

**Teacher note:** students will come to this focus area from diverse backgrounds, experience and understanding. There is no prerequisite mandated for this course and students may not have studied the Computing Technology 7–10 syllabus. A pre-test will enable teachers to determine a baseline measurement of understanding, track growth in understanding and help determine next steps. Teachers could activate prior knowledge using [KWLH charts](https://app.education.nsw.gov.au/digital-learning-selector). These are explicit teaching strategies.

1. Are the internet and the web the same thing? Explain in the space below.

|  |
| --- |
| **Sample answer:**  The terms ‘internet’ and ‘web’ are related, though they are not the same thing. The internet is the global network that connects devices, while the web is a collection of interconnected webpages accessible through browsers.  The internet is the global network of interconnected computers and devices. The web is a subset of the internet consisting of interconnected hypertext documents and resources accessed via web browsers.  The web relies on the internet's infrastructure to function, but it represents a specific subset of content and services available on the internet. |

1. What is HTML and what is its main role in web development? Explain in the space below.

|  |
| --- |
| **Sample answer:**  HTML stands for hypertext markup language. Its primary purpose in web development is to structure the content of webpages by defining the various elements such as headings, paragraphs, links, images and so on. It defines how these elements will be displayed through the browser. |

1. Describe the role of CSS in web development and provide an example of how it enhances the presentation of webpages. Explain in the space below.

|  |
| --- |
| **Sample answer:**  CSS (cascading style sheets) is used to style the HTML elements and control their appearance on webpages. For example, CSS can be used to set the colour, font size, margin, padding and layout of elements. An example would be styling a `<div>` element with CSS to give it a specific background colour, font size and border. |

1. What is JavaScript, and what are its main functions in web programming? Explain in the space below.

|  |
| --- |
| **Sample answer:**  JavaScript is a programming language commonly used in web development. Its main functions include adding interactivity to webpages, manipulating HTML and CSS, handling user input, making asynchronous requests to servers |

1. Explain the difference between client-side and server-side programming in web development. Explain in the space below.

|  |
| --- |
| **Sample answer:**  Client-side programming refers to code executed in the user's web browser, such as JavaScript, which handles interactions and modifications of the user interface. Server-side programming refers to code executed on the web server, handling tasks like processing form submissions, querying databases and generating dynamic content. |

1. What is the purpose of a web browser's developer tools and how can they assist in web development?

|  |
| --- |
| **Sample answer:**  The purpose of web browser developer tools is to assist developers in debugging, testing and optimising webpages. They provide features such as inspecting HTML/CSS, debugging JavaScript, analysing network requests and optimising page performance. |

1. What is a web server and what role does it play in serving webpages to users?

|  |
| --- |
| **Sample answer:**  A web server is software that stores, processes and delivers webpages to users over the internet. It responds to requests from web browsers by sending the requested webpages, typically using HTTP or HTTPS protocols. |

1. Explain the concept of cookies and how they are used in web development.

|  |
| --- |
| **Sample answer:**  Cookies are small pieces of data stored on a user's device by websites to remember user-specific information, such as login credentials, preferences and shopping cart items. They are commonly used in web development for session management, personalisation and tracking user behaviour. |

1. What are some common security considerations in web development and how can developers reduce these risks?

|  |
| --- |
| **Sample answer:**  Common security considerations in web development include protecting against cross-site scripting (XSS), SQL injection, cross-site request forgery (CSRF) and ensuring secure communication over HTTPS. Developers can mitigate these risks by validating and sanitizing user input, implementing proper authentication and authorization mechanisms, using HTTPS to encrypt data in transit and staying informed about security best practices and updates. |

**Teacher note:** security considerations will be addressed in detail in the Secure software architecture focus area.

1. Web developers often refer to:

* HTML as Structure
* CSS as Style
* JavaScript as Action.

Explain this in the space below.

|  |
| --- |
| **Sample answer:**  HTML (hypertext markup language) as Structure: HTML is the backbone of webpages. It provides the basic structure and content of a webpage by defining elements such as headings, paragraphs, lists, images, links and more. HTML organises the content in a structured way, forming the foundation on which the webpage is built. It determines what content appears on the page and how it is organised, but it doesn't dictate the visual appearance or behaviour of the elements.  CSS (cascading style sheets) as Style: CSS is used to style the presentation of HTML elements. It controls the visual appearance of the content defined by HTML, such as colours, fonts, spacing, layout and more. With CSS, web developers can customise the look and feel of a webpage, making it visually appealing and user-friendly. CSS allows for separation of content and presentation, enabling developers to create consistent and visually appealing designs across multiple webpages.  JavaScript as Action: JavaScript is a programming language that adds interactivity and dynamic behaviour to webpages. While HTML provides the structure and CSS provides the style, JavaScript adds functionality and behaviour to elements on the page. With JavaScript, developers can create interactive features such as form validation, animations, user interface enhancements, dynamic content updates and much more. JavaScript enables webpages to respond to user actions, events and inputs, making the web experience more engaging and interactive. |

For students needing further support, assign the video below as homework:

[Learn HTML -– Your First Web Page (For Absolute Beginners) (4:33)](https://youtu.be/BBcq9abR9Es?si=54FpAEUBkJiBDkWz)

[Learn CSS (Tutorial) -– Basics + Selectors (For Absolute Beginners) (9:51)](https://youtu.be/Awmx8XlgcaE?si=0S0SMCGf5IWHGUx5)

**Teacher note:** some Software engineering teachers may approach their TSO’s or computer co-ordinators for an old laptop to recycle into a web server. A desktop or raspberry pi could also be used. This web server would ensure delivery of the syllabus content in [Programming for the web](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/content/n12/fa6aab137e) is done with an authentic, engaging and project based approach and could be used as a foundation for the Year 12 Project.

This unit including the assessment task **does not** require this extra hardware and the Express framework for Node.js within the VSCode IDE to create a PWA (Appendix 1).

The learning experience through establishing this will inform students considering extending on the Programming for the web focus area for their major project. They will deepen their knowledge of the ‘plumbing’ and coding required at both front and back ends of the web.

Please refer to the Web server guide in [Appendix 2 – web server guide](#_Appendix_2:_Web) of this document.

## Data transmission using the web

**Teacher note:** the following video could be issued as homework as part of a flipped classroom model where students discuss the contents during class.

* [What is HTTP? How the Internet Works! #1 (17:25)](https://youtu.be/wW2A5SZ3GkI?si=PRPeLqcB9nC-7F_S)

### Explore the applications of web programming

**Including: interactive website/webpages, e-commerce, progressive web apps (PWAs)**

**Activity 1:** [Think-Pair-Share](https://app.education.nsw.gov.au/digital-learning-selector/LearningActivity/Card/645?clearCache=befbfa2e-bd65-40ce-da3f-b12bf11851b5) a list of all the applications of web programming. Include a brief description of each application in the space below.

|  |
| --- |
| **Sample answer:**  **E-commerce websites:** online stores and marketplaces where users can browse products, add items to their cart and make purchases securely.  **Social media platforms:** websites and applications that allow users to connect, share content, communicate and interact with others online.  **Content management systems** **(CMS**): platforms for creating, managing and publishing digital content, such as blogs, news websites and corporate intranets.  **Online learning platforms:** websites and applications for delivering educational content, courses, tutorials and interactive lessons to users.  **Web-based games:** browser-based games and gaming platforms that allow users to play games directly in their web browsers without the need for downloads or installations.  **Project management tools:** web-based platforms for organising tasks, collaborating with team members, tracking progress and managing projects.  **Financial services:** online banking platforms, payment gateways and financial management tools for managing transactions, accounts and investments.  **Healthcare applications:** web-based healthcare portals for managing patient information and appointments.  **Real estate portals:** websites and applications for listing properties, searching for homes and connecting buyers, sellers and agents in the real estate market.  **Weather forecasting services:** web applications that provide weather forecasts, radar images and meteorological data to users.  **Job portals and recruitment platforms:** websites for posting job listings, searching for employment opportunities and connecting job seekers with employers.  **Government and public services:** websites for accessing government information, submitting forms, paying taxes and accessing public services online. |

#### What does ‘progressive’ mean in progressive web apps?

**Activity 2:** read the following to a classmate and ask them to provide a summary of what you have read.

PWAs are built with progressive enhancements. This means they're designed to work for every user regardless of the chosen browser They start with a basic version that works on all browsers and progressively enhance the experience on modern browsers.

They are designed to be responsive. Responsiveness has nothing to do with loading speed in this context rather it means they adapt to various screen sizes and devices, providing a consistent user experience across different platforms.

PWAs are designed to function even when there's limited or no internet connectivity, using service workers to cache content and provide offline functionality.

They offer an app-like experience to users, including features such as push notifications, installation to the home screen and full-screen mode, making them feel like mobile apps.

PWAs are always up to date because they are web-based and do not require manual updates. Users always get the latest version when they access the app.

|  |
| --- |
| **Sample answer:**  PWAs progressively enhance user experiences across different devices and network conditions, while also providing features typically associated with mobile apps. |

#### Recognising a PWA

**Installation prompt**

If you visit a website that is a PWA, some browsers, especially on mobile devices, may prompt you to install the app to your home screen. This typically happens if you visit the site frequently.

**Home screen icon after installation**

If the PWA is added to your home screen, it will have an icon just like any other app. If you see an app icon for a website you frequently visit, it's likely a PWA.

**Full-screen mode**

PWAs often offer a full-screen mode, which hides the browser's URL bar and navigation controls, giving you an experience similar to an app.

**Offline functionality**

If you notice that the app still works even when you're offline or have a spotty internet connection, it's likely a PWA. PWAs use service workers to cache content and provide offline functionality.

**Push notifications**

PWAs can send push notifications, similar to native mobile apps. If you receive push notifications from a website, it might be a PWA.

**Responsive design**

PWAs are built to be responsive, meaning they adapt to various screen sizes and devices. If the app adjusts well to different screen sizes, it could be a PWA.

**Check browser settings**

Some browsers have specific settings or developer tools that allow you to see if a website is a PWA. For example, in Google Chrome, you can go to **Settings** > **Site Settings** > **Installed Apps** to see a list of installed PWAs.

**Not all websites have PWAs**

This means, if none of the above indicators are evident it is unlikely to be a PWA.

**Activity 3:** **homework** – listen to the podcast: *The Machine: A computer science education* ‘[Beginner's Guide To Web Development](https://podfollow.com/the-machine/episode/0be5037a4b40d3d1c86d4e5a8e89178503162a51/view)’.

[Think-Pair-Share](https://app.education.nsw.gov.au/digital-learning-selector/LearningActivity/Card/645?clearCache=befbfa2e-bd65-40ce-da3f-b12bf11851b5) a list of project ideas for your PWA.

|  |
| --- |
| **Sample answer:**  Suggested topics to catalogue include:  **videos, songs, websites, actors, games, sports, restaurants, food, recipes, art, pop culture, quotes, canteen items, clothes,** m**ovies.**  **Music example with fields**  **music,** Title ( The name of the track or album), Artist (The name of the artist or band who performed the track), Album(The name of the album the track belongs to), Genre (The genre or category of music e.g., rock, pop, hip-hop)., Release Date (The date when the track or album was released), Track Number( The position of the track on the album), Duration( The length of the track in minutes and seconds). |

Think-Pair-Share a list of project ideas for your PWA.

**Activity 4:** select one topic or suggest another and provide a description of how it would work from the user’s perspective. Include the data that may be catalogued. A sample is provided.

### Investigate and practise how data is transferred on the internet

**Including: data packets, internet protocol (IP) addresses, including IPv4, domain name systems (DNS)**

**Activity 5:** watch the video [How does the internet work? (8:58)](https://youtu.be/x3c1ih2NJEg?si=p6OFdBPjdd7vfR_-).

A brief overview of how data travels across the internet from one point to another is best introduced via explicit teaching.

The teacher summarises the video by narrating while drawing on the whiteboard (think-alouds), the earth with data packets being sent independently across a network taking different routes and reassembled at the destination into the original data.

#### Understanding data packets

**Activity 6:** watch the video [A Packet's Tale. How Does the Internet Work? (3:29)](https://youtu.be/ewrBalT_eBM?si=8UY4mc-W7Eg1It25). In the space below describe data packets and why they are fundamental to the internet.

|  |
| --- |
| **Sample answer:**  Data packets are small units of data transmitted over a network. They have a header (containing source and destination IP addresses), the actual data (information or message) and a footer. Data packets ensure efficient and reliable data transfer. |

#### Understanding Internet Protocol (IP) addresses

**Activity 7:** watch the video [IP addresses explained (3:32)](https://youtu.be/7_-qWlvQQtY?si=6imRoAVMVexMk4Q3).

In the space below describe IP addresses and IPv4.

|  |
| --- |
| **Sample answer:**  IPv4 (Internet Protocol version 4) addresses are used to uniquely identify devices on a network. The format consists of 4 decimal numbers separated by periods, each ranging from 0 to 255. Each of these numbers represents an 8-bit octet, totalling 32 bits for the entire address.  For example, an IPv4 address might look like this: 192.168.0.1 where:   * **192** represents the first octet * **168** represents the second octet * **0** represents the third octet * **1** represents the fourth octet.   When written out in binary, each of these decimal numbers is represented by an 8-bit binary sequence. For example, the number 192 in binary is 11000000, and 1 in binary is 00000001.  IPv4 addresses are used for communication between devices on the internet. Each device connected to the internet must have a unique IPv4 address to ensure that data packets are routed correctly. However, due to the limited number of possible combinations with IPv4 (approximately 4.3 billion unique addresses), IPv6 was developed to accommodate the growing number of devices connected to the internet. |

**Activity 8:** **homework** – watch [IP Addresses Explained (12:50)](https://youtu.be/LIzTo6e4FgY?si=0JowzuotyHtvMkoH) and take notes to consolidate your understanding.

#### Domain Name Systems (DNS)

**Activity 9:** watch the video [DNS Explained in 100 Seconds (2:15)](https://youtu.be/UVR9lhUGAyU?si=teH2aGUWDXoBRnjH).

In the space below list the steps in browsing the internet to explain how the DNS works

|  |
| --- |
| **Sample answer:**  The Domain Name System (DNS) serves as the internet's phone book, translating human-readable domain names (like ‘example.com’) into IP addresses that computers use to identify each other on the network.  **Steps**   1. User input: a user enters a domain name (like ‘example.com’) into their web browser. 2. DNS lookup: the browser sends a DNS query to a DNS resolver, typically provided by the user's internet service provider (ISP) or configured in their network settings. 3. DNS resolver: the DNS resolver receives the query and checks its cache to see if it already has the IP address for the requested domain. If the address is found in the cache and is still valid (has not expired), the resolver returns it immediately to the browser. If not, it proceeds to step 4. 4. Recursive query: if the resolver doesn't have the IP address in its cache, it becomes responsible for finding it. It starts by querying the root DNS servers, which hold information about the top-level domains (like ‘.com’, ‘.org’, ‘.net’ and so on). 5. Top-level domain (TLD) server: the root server directs the resolver to the appropriate TLD server based on the domain extension (for example, ‘.com’). The TLD server holds information about the authoritative name servers responsible for the next level of the domain hierarchy. 6. Authoritative name server: the resolver contacts the authoritative name server for the specific domain (for example, ‘example.com’). This server holds the most up-to-date information about the domain, including its IP address. 7. Response: the authoritative name server responds to the resolver with the requested IP address. 8. Cache update: the resolver caches the IP address for future use, reducing the need for repeated lookups. 9. Browser request: armed with the IP address obtained from DNS, the browser can now send a request to the appropriate server hosting the website. 10. Website access: the server responds to the browser's request and the website is displayed to the user.   This process allows users to access websites using easy-to-remember domain names while leveraging the underlying IP addressing system of the internet for communication between devices. DNS operates as a distributed system, with multiple levels of servers working together to efficiently resolve domain names to IP addresses.  DNS is a system that translates human-readable domain names into IP addresses.  DNS servers resolve domain names to IP addresses.  DNS facilitates user-friendly web browsing experiences. |

**Activity 10:** **homework** – students consolidate understanding by listening to the podcast ‘[How The Internet Works](https://podfollow.com/the-machine/episode/a513be7ecc8ddcabbdc08a6578b0dce6c55bfc47/view)’ and taking notes.

#### Practical exercise: tracing data transfer

**Activity 11:** watch the video [Traceroute (tracert) Explained – Network Troubleshooting (9:23)](https://youtu.be/up3bcBLZS74?si=cK3nMwZlM4fCZtOH).

Students compare results from traceroute and ping commands using command prompt or terminal to trace the path of data packets from their device to a specific website.

Students analyse the output to identify the IP addresses of intermediate routers and the final destination server.

Perform a tracert on your computer and take a screen shot. Annotate the screen shot to explain how the path data packets travel can be traced.

|  |
| --- |
| **Sample answer:**  Tracert (traceroute) is a command-line tool used to trace the path that data packets take from your computer to a destination IP address or domain name. Tracert shows the network hops (routers) between your computer and the destination.  The first line of the tracert output shows the IP address of the first router or gateway the computer communicates with. This IP address is usually the default gateway of the local network or ISP's gateway.  Tracert might perform a reverse DNS lookup on the IP addresses of each hop to display their domain names, if available. This reverse lookup converts the IP addresses back into domain names. However, not all routers may have reverse DNS entries, so you might see IP addresses instead of domain names.  As tracert progresses, it displays the IP addresses of intermediate routers along the path to the destination. If reverse DNS lookup is successful, you might see domain names instead of IP addresses for some of these hops.  Tracert eventually reaches the destination IP address or domain name, showing its IP address and, if available, the domain name associated with it. |

### Investigate and describe the function of web protocols and their ports

**Including: HTTP, HTTPS, TCP/IP, DNS, FTP, SFTP, SSL, TLS, SMTP, POP 3, IMAP**

**Activity 12:** [Think-Pair-Share](https://app.education.nsw.gov.au/digital-learning-selector/LearningActivity/Card/645?clearCache=befbfa2e-bd65-40ce-da3f-b12bf11851b5) the reasons software engineers need to understand web protocols and ports when programming for the web. Describe these reasons in the space below.

|  |
| --- |
| **Sample answer:**  Designing and developing applications that communicate with other systems and services over the internet. Knowing how data is transmitted and received, software engineers can optimise their applications for speed, reliability and security.  Diagnosing and troubleshooting problems effectively. They can analyse network traffic, identify bottlenecks and resolve connection issues.  Implementing secure communication practices, such as encryption, authentication and data integrity verification, in their applications. This helps protect sensitive user data from interception and manipulation by malicious actors.  Ensuring compatibility and interoperability between different components of the application stack. Software engineers can develop interfaces and integration points that meet standard protocols, providing seamless communication between different systems.  Optimising the performance of their web applications. By understanding how data is transmitted over the network, software engineers can minimise latency and maximise bandwidth utilisation. This optimisation enhances the responsiveness and scalability of web applications, providing a better user experience.  Complying with industry standards and best practices and adhere to these standards when developing web applications to ensure compatibility with existing infrastructure and compliance with regulatory requirements. Understanding web protocols helps software engineers implement standards-compliant solutions that adhere to industry norms and guidelines. |

**Teacher note:** security considerations will be addressed in detail in the Secure software architecture focus area.

Web protocols are essential for facilitating communication between devices over the internet. Each protocol serves a specific purpose and operates on designated ports within the TCP/IP networking model. Understanding these protocols and their associated ports is crucial for managing network communication and ensuring secure and efficient data transfer over the internet.

**Teacher note:** use [connected learning strategies](https://education.nsw.gov.au/teaching-and-learning/curriculum/explicit-teaching/explicit-teaching-strategies/connecting-learning) to build on students understanding. Analogies can be used to build on familiar pre-existing knowledge. These analogies should be corrected and fine-tuned as understanding grows.

**Activity 13:** imagine you are sending an (‘old school’) letter to a friend. Think of web protocols as the rules and guidelines for sending and receiving these letters.

Just like there are specific rules for addressing envelopes, postage and delivery, web protocols define how data is packaged, transmitted and received over the internet.

Ports are like different mailboxes at the post office. Each mailbox (port) is assigned a specific number, and different types of mail (data) go to different mailboxes. For example, letters might go to one mailbox, packages to another, and urgent mail to yet another. Similarly, on the internet, different types of data traffic use specific ports to ensure it goes to the right destination.

HTTP (Port 80): imagine HTTP as the standard mailbox at the post office. It's used for sending and receiving regular letters (webpages) between web browsers and servers. HTTPS is like a special mailbox with a lock. It's used for sending sensitive letters (secure webpages) that need to be protected from prying eyes. This mailbox ensures that the contents of the letters (data) are encrypted for security. HTTPS stands for Hypertext Transfer Protocol Secure and is a secure version of HTTP,( the protocol used for transferring data between a web browser and a website). When you visit a website that uses HTTPS, the data sent between your browser and the website is encrypted, which helps to protect it from being intercepted by hackers.

Port 443 is the port number that is commonly used for HTTPS communication. Ports are like virtual doors that allow different types of data to pass through a network. When you access a website using HTTPS, your browser communicates with the website's server through port 443 to establish a secure connection. HTTPS and Port 443 work together to ensure that the data you send and receive while browsing the internet is encrypted and secure, helping to protect your privacy and sensitive information.

DNS (Port 53): DNS is like a directory service at the post office. When you write your friend's name on an envelope, the post office looks up their address in the directory to know where to deliver the letter. Similarly, DNS translates domain names (like "example.com") into IP addresses so that data packets know where to go on the internet.

FTP (Port 20, 21): FTP is like a special mailbox and package pickup area at the post office. It's used for sending and receiving large files (packages) between computers, with one mailbox for sending and one for receiving.

SMTP (Port 25): SMTP is like a dedicated mailbox for sending letters (emails) from one mail server to another. It ensures that emails are delivered reliably and efficiently across different mail servers.

POP3 (Port 110) and IMAP (Port 143): POP3 and IMAP are like personal mailboxes at the post office. They allow you to access your letters (emails) stored on the mail server, either by downloading them (POP3) or by viewing them remotely (IMAP).

SSL/TLS (Various Ports): SSL/TLS are like special security envelopes that you use for important letters. They encrypt the contents of the letter (data) to ensure that nobody can read it while it's in transit.

**Activity 14:** research and complete the table of some common web protocols, their functions and associated ports.

|  |  |  |
| --- | --- | --- |
| Protocol | Function | Port |
| HTTP (Hypertext Transfer Protocol) | The foundation of data communication on the world wide web. It allows web browsers to retrieve webpages from web servers. | Default port is 80. |
| HTTPS | The secure version of HTTP, adding encryption using SSL/TLS to ensure data privacy and integrity during transmission. | Default port is 443. |
| TCP/IP (Transmission Control Protocol/Internet Protocol) | A suite of communication protocols that enable network communication between devices on the internet. It provides end-to-end data transmission. | Various protocols within the TCP/IP suite use different ports. For example, HTTP (80), HTTPS (443), DNS (53), FTP (20, 21), SMTP (25), etc. |
| DNS (Domain Name System) | Translates domain names into IP addresses, allowing users to access websites using human-readable domain names instead of numerical IP addresses. | Default port is 53. |
| FTP (File Transfer Protocol) | Is used for transferring files between a client and a server on a computer network. It provides a way to upload, download and manage files. | Uses ports 20 (data transfer) and 21 (control). |
| SFTP (SSH File Transfer Protocol) | A secure file transfer protocol that provides file access, transfer, and management over a secure data stream. It operates over SSH (Secure Shell) protocol. | Default port is 22. |
| SSL (Secure Sockets Layer) and TLS (Transport Layer Security) | SSL and its successor TLS are cryptographic protocols that provide secure communication over a computer network. They are commonly used to secure data transmission in HTTPS, SMTP and other protocols. | Do not have specific ports but are often used with protocols like HTTPS (443), SMTP (25, 587), and IMAP (993). |
| SMTP (Simple Mail Transfer Protocol) | Used for sending email messages between servers. It defines how email messages should be relayed between servers over the internet. | Default port is 25. |
| POP3 (Post Office Protocol version 3) and IMAP (Internet Message Access Protocol) | Email retrieval protocols used by email clients to retrieve emails from a mail server. | POP3 uses port 110, while IMAP typically uses port 143. SSL/TLS versions of these protocols (POP3S and IMAPS) use ports 995 and 993, respectively. |

### Explain the processes for securing the web

**Including: Secure Sockets Layer (SSL) certificates, encryption algorithms, encryption keys, plain text and cipher text, authentication and authorisation, hash values, digital signatures**

**Teacher note:** the following video could be issued to watch as homework as part of a flipped classroom strategy [Web Application Security Fundamentals (26:24)](https://youtu.be/-7OX58nHPb8?si=kV7Vz0kEKHXKgHaB).

Securing the web involves implementing various technologies and practices to protect data transmission, authenticate users and servers, and ensure privacy and integrity of information.

**Activity 15:** [Jigsaw activity](https://app.education.nsw.gov.au/digital-learning-selector/) – ‘Securing the web’.

Jigsaw is a cooperative learning strategy that:

* supports educators to differentiate learning
* enables each student to specialise in one aspect of a topic, with each student's part being essential to the completion of the task
* builds students' comprehension, cooperation, communication and problem-solving skills.

Assign students into groups numbered 1 to 7.

* Students numbered 1 form an expert group that will research Secure Sockets Layer (SSL) certificates.
* Students numbered 2 form an expert group that will research encryption algorithms.
* Students numbered 3 form an expert group that will research encryption keys.
* Students numbered 4 form an expert group that will research plain text and cipher text.
* Students numbered 5 form an expert group that will research authentication and authorisation.
* Students numbered 6 form an expert group that will research hash values.
* Students numbered 7 form an expert group that will research digital signatures.

Each group completes the research task and creates 2 slides for a class collaborative PowerPoint or presentation about securing the web process they have researched.

The completed presentation is presented to the class and students complete the following table.

(Sample answers are provided with the minimum detail required.)

|  |  |
| --- | --- |
| Securing the web | Explanation |
| 1. Secure Sockets Layer (SSL) certificates | SSL certificates are digital certificates that authenticate the identity of a website and encrypt data transmitted between the web server and the user's browser. These certificates are issued by trusted Certificate Authorities (CAs) and contain information such as the domain name, public key, expiration date and the digital signature of the CA.SSL certificates enable HTTPS, ensuring that data transmitted over the internet is encrypted and secure from eavesdropping and tampering. |
| 1. Encryption algorithms | Encryption algorithms are mathematical formulas used to convert plain text into cipher text, making it unreadable without the proper decryption key. Common encryption algorithms include AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and DES (Data Encryption Standard).These algorithms use keys to encrypt and decrypt data, ensuring confidentiality and privacy. |
| 1. Encryption keys | Encryption keys are used to encrypt and decrypt data. There are 2 types of keys: public keys and private keys. Public keys are used for encryption and are freely distributed, while private keys are kept secret and used for decryption .In asymmetric encryption, data encrypted with a public key can only be decrypted with the corresponding private key, providing secure communication between parties. |
| 1. Plain text and cipher text | Plain text refers to readable data in its original form. Cipher text refers to data that has been encrypted using an encryption algorithm and is unreadable without the decryption key. |
| 1. Authentication and authorisation | Authentication verifies the identity of users or servers accessing a system or service. It ensures that only authorised individuals or entities can access sensitive information.  Authorisation determines the level of access granted to authenticated users. It defines what actions users are allowed to perform within a system or application. |
| 1. Hash values | Hash values are unique identifiers generated by applying a hash function to data. They represent a fixed-size string of characters that uniquely identifies the input data. Hash functions are one-way functions, meaning that it is computationally infeasible to reverse the process and obtain the original data from the hash value. Hash values are commonly used for data integrity verification, password hashing and digital signatures. |
| 1. Digital signatures | Digital signatures are cryptographic mechanisms used to authenticate the origin and integrity of digital documents or messages. They involve the use of public key cryptography, where the sender signs the document with their private key, and the recipient verifies the signature using the sender's public key. Digital signatures provide non-repudiation, ensuring that the sender cannot deny having sent the document, and data integrity, ensuring that the document has not been altered since it was signed. |

**Summary**: by implementing SSL certificates, encryption algorithms, encryption keys, authentication and authorisation mechanisms, hash values, and digital signatures, web applications can secure data transmission, protect sensitive information, and authenticate users and servers, to ensure a safe and trustworthy online environment.

### Investigate the effect of big data on web architecture

**Including: data mining, metadata, streaming service management**

#### What is big data?

Big data refers to extremely large and complex datasets that cannot be easily managed, processed or analysed using traditional data processing tools or methods.

The term ‘big data’ encompasses not only the volume of data but also its velocity, variety, veracity and value.

**Activity 16:** use the following words to complete the cloze passage on definitions.

**Keywords:** diverse, convergence, value, volumes, velocity, challenges, varying, opportunities

Big data represents the [1] \_\_\_\_\_\_\_\_\_\_ of large [2] \_\_\_\_\_\_\_\_\_\_ of data, high [3] \_\_\_\_\_\_\_\_\_\_ of data generation, [4] \_\_\_\_\_\_\_\_\_\_ data types, [5] \_\_\_\_\_\_\_\_\_\_ data quality, and the potential to derive [6] \_\_\_\_\_\_\_\_\_\_ from data analytics. It poses both [7] \_\_\_\_\_\_\_\_\_\_ and [8] \_\_\_\_\_\_\_\_\_\_ for organisations seeking to harness the power of data to gain competitive advantage and drive business success.

#### What is web architecture?

Web architecture refers to the structure, components, and principles underlying the design and workings of web-based systems and applications. It includes the software, hardware, protocols, standards, and technologies that enable the functioning of websites, web services and web applications.

Key components of web architecture include:

**1. Client-side components**

1.1 Web browsers: software applications that retrieve, render, and display web content to users.

1.2 Client-side scripting languages (for example, JavaScript): used to enhance interactivity and functionality within webpages.

1.3 User interfaces (UI): design elements and layouts that enable users to interact with web applications.

**2. Server-side components**

2.1 Web servers: software programs that host and serve web content in response to client requests.

2.2 Application servers: middleware that provides runtime environments for executing server-side scripts and applications.

2.3 Databases: systems for storing and managing structured data used by web applications.

**3. Networking infrastructure**

3.1 Internet protocols (for example, HTTP, HTTPS, TCP/IP): standards governing communication between clients and servers over the internet.

3.2 Domain Name System (DNS): translates domain names into IP addresses, facilitating web address resolution.

3.3 Content delivery networks (CDNs): distributed networks of servers that deliver web content to users based on their geographic location, improving performance and reliability.

**4. Data formats and protocols**

4.1 Hypertext markup language (HTML), cascading style sheets (CSS), JavaScript (JS): standard languages for creating and styling web content.

4.2 Representational state transfer (REST), Simple object access protocol (SOAP): protocols for exchanging data between web services and clients.

4.3 XML, JSON: data interchange formats commonly used for transmitting structured data over the web.

**5. Security mechanisms**

5.1 Secure Sockets Layer (SSL)/Transport Layer Security (TLS): protocols for encrypting data transmitted between clients and servers, ensuring confidentiality and integrity.

5.2 Authentication and authorisation mechanisms: techniques for verifying the identity of users and controlling access to web resources.

5.3 Firewalls, intrusion detection systems (IDS), and other security measures: protect web applications from unauthorised access, attacks and data breaches.

**Activity 17:** students form teams of 3.

**Scenario:** your team have been requested to design the graphics for a new software engineering textbook. In the space provided on the following page sketch a diagram to represent web architecture that includes all key components.

* The diagram should indicate how each component is related to the others.
* Share this diagram with the class for peer assessment.
* Compare the diagram with online depictions and explain any differences.
* Teams modify and improve the diagrams by using Lucidchart or flowcharts and icons (Word) to create a finished poster.

|  |
| --- |
|  |

#### The effect of big data on web architecture

Big data has significantly impacted web architecture, particularly in terms of handling and processing large volumes of data efficiently. This has involved: data mining, metadata and streaming service management. Big data has led to the evolution of web architectures towards more scalable, flexible and real-time data processing systems.

**Activity 18:** [Jigsaw activity](https://app.education.nsw.gov.au/digital-learning-selector/) – ‘The effect of big data on web architecture’

Jigsaw is a cooperative learning strategy that enables concepts to be covered efficiently. Most successful jigsaw activities arrange groups of students randomly and differently from past groups. Working with different people is an important skill for any future pathway.

Assign students into home groups (A, B, C, and so on) of 3.

Number each student 1, 2 or 3.

* Students numbered 1 form into an expert group that will research data mining.
* Students numbered 2 form into an expert group that will research metadata.
* Students numbered 3 form into an expert group that will research streaming service management.

Each group research their topic in relation to ‘the effect of big data on web architecture’.

After 20 minutes of research and discussion on their topic students return to their homegroup (A, B, C) to peer tutor the other members about their research topic.

Each group completes the description table and then swaps with another group for peer assessment according to the sample response provided.

|  |  |
| --- | --- |
| Big data | Explanation of big data on architecture |
| Data mining | Big data technology enables organisations to collect, store, and analyse large amounts of data from diverse sources. Data mining is used to extract valuable insights, patterns and trends from these massive datasets, helping businesses make data-driven decisions. Web architectures have evolved to incorporate scalable data storage systems, distributed processing frameworks (for example, Hadoop and Spark), and machine learning algorithms to support data mining activities. |
| Metadata | Metadata plays a crucial role in big data systems, providing context and structure to large datasets. In web architectures, metadata is used to categorise, organise and manage vast amounts of web content, making it easier to search, discover and analyse. |
| Streaming service management | With real-time data processing and streaming analytics, web architectures have adapted to support streaming service management. Streaming platforms enable organisations to ingest, process and analyse continuous streams of data in real-time, allowing for instant insights and decision-making. |

**Activity 19:** **extension homework** – watch [Understanding Big Data for Software Engineers (58:28)](https://youtu.be/NXjSJYyByzE?si=LV0f3reV2s51iqKV).

## Designing web applications

**Activity 20:** watch [Tim Berners-Lee: How This Guy Invented the World Wide Web 30 Years Ago (5:56)](https://youtu.be/GUrDI6OkJfU?si=yiVVhnwjfDXeXHEt).

### Investigate and explain the role of the World Wide Web Consortium [(W3C)](https://www.w3.org/) in the development of applications for the web.

**Including: Web Accessibility Initiative (WAI), internationalisation, web security, privacy, machine-readable data**

Students building a PWA should be introduced to the standards and guidelines provided by W3C. This provides insight into the development process and may improve their application. Students investigate and explore how each of the following aspects contributes to the development of their PWA.

#### Web Accessibility Initiative (WAI)

Students building PWAs will benefit from WAI guidelines by ensuring that they are accessible to users with disabilities. Implementing WCAG recommendations ensures that PWAs are perceivable, operable, understandable and robust for all users, including those with visual, auditory, physical, speech, cognitive and neurological disabilities. Accessible PWAs improve usability and inclusivity, leading to a better user experience for all users.

**Activity 21:** [Jigsaw activity](https://app.education.nsw.gov.au/digital-learning-selector/) – ‘Web Accessibility’

Assign students into home groups (A , B, C, and so on) of 3.

Number each student 1, 2 or 3

* Students numbered 1 form into an expert group that will create a webpage on:
* [Writing for Web Accessibility](https://www.w3.org/WAI/tips/writing/)
* Students numbered 1 form into an expert group that will create a webpage on:
* [Designing for Web Accessibility](https://www.w3.org/WAI/tips/designing/)
* Students numbered 1 form into an expert group that will create a webpage on:
* [Developing for Web Accessibility](https://www.w3.org/WAI/tips/developing/)

Each group creates a webpage using HTML on the tips they have researched. They use snippets of code samples suggested by the [tutorials](https://www.w3.org/WAI/tutorials/).

Menu for the Web accessibility tutorials.


Image sourced from [Tutorials](https://www.w3.org/WAI/tutorials/) by [W3C](https://www.w3.org).

Each member of the expert group returns to their home group to share their findings and webpage.

#### Internationalisation

W3C's [Internationalization (I18n)](https://www.w3.org/International/) standards help developers create PWAs that support multilingual content and accommodate diverse cultural conventions. By following I18n guidelines, developers can ensure that PWAs are localised and adapted to different languages and regions, catering to a global audience. This enhances the accessibility and usability of PWAs for users worldwide.

**Activity 22:** try out the [Internationalization Checker](https://validator.w3.org/i18n-checker/) on a number of different websites you are familiar with by pasting their URLs into the checker.

Take a screen shot of the results for any that are flagged.

|  |
| --- |
| Internationalization Checker results |

#### Web security

Security is crucial for PWAs to protect users' data and ensure safe browsing experiences. W3C's standards for web security, such as HTTPS, CSP, CORS and secure authentication mechanisms, help developers build secure PWAs that mitigate common security threats, including XSS, CSRF and injection attacks. By adhering to W3C's security standards, developers can instil trust and confidence in their PWAs This encourages the use of the PWA.

**Teacher note:** security considerations will be addressed in detail in the Secure software architecture focus area.

#### Privacy

Privacy is a concern for all software engineers. This is especially the case when developing PWAs because of the sensitive data they handle, such as user preferences, location information and personal data.

W3C has privacy standards and guidelines that developers can follow to design PWAs that respect users' privacy rights and comply with regulatory requirements. Software engineers can include features like browser privacy controls, user consent mechanisms and secure data handling practices to build PWAs that prioritize user trust and privacy protection.

#### Machine-readable data

Developers building PWAs can use W3C standards for representing and exchanging machine-readable data. This enhances interoperability and automation by making it easier to understand and process by search engines, web crawlers and other applications. Being machine readable makes the PWAs content easier to discover and access. This makes it more visible and effective on the web.

**Activity 23:** in the space below summarise the role of the World Wide Web Consortium [(W3C)](https://www.w3.org/) in the development of applications for the web.

|  |
| --- |
| **Sample answer:**  In summary, by following W3C standards and guidelines developers and software engineers can ensure that the PWAs are accessible, inclusive, secure, respect privacy and are interoperable.  The W3C provides best practices for developers creating high-quality PWAs that deliver improved user experiences while promoting trust, privacy and innovation on the web. |

### Model elements that form a web development system

**Including: client-side (front-end) web programming, server-side (back-end) web programming, interfacing with databases that are based on Structured Query Language (SQL) or non-SQL**

**Teacher note:** the modelling of these elements introduces students to key concepts they require to create their PWA and complete the assessment task. These models should be developed into prototypes or working models that can be used during the development of their assessment task.

**Activity 24:** students visit [Web application Development & Testing Made Easy](https://www.browserstack.com/guide/web-application-development-guide).

Read the sections:

* What is Web Application Development?
* Web Application Development Process.

Using the space below draw a diagram that represents the web development system.

|  |
| --- |
|  |

Compare your diagram with [an online example](https://medium.com/weekly-webtips/everything-you-need-to-know-about-web-application-architecture-57d94ff44c1b).

Students considering Programming for the web as the foundation for their major project are encouraged to use extension material such as: [Harvard CS50’s Web Programming with Python and JavaScript (14:14:23)](https://youtu.be/vzGllw18DkA?si=Lr25walMPEOHiqbQ).

#### Client-side (front-end) web programming

Students plan design and produce theclient-side (front-end) of their PWA that catalogues a personal interest. Remember during the design that the application should also allow users to query the catalogue. Students’ personal interest catalogues could include: music, movies, videos, songs, websites, actors, games, restaurants, food, recipes, art, pop culture, quotes, canteen items, clothes and so on.

**Activity 25:** students create the user interface (UI) to interact with directly in their web browsers including:

* design and layout the UI by creating a wireframe or storyboard for the pages that will be present in the system
* evidence of accessibility and inclusivity considerations
* evidence of usability considerations
* technologies:
* HTML (for structure)
* CSS (for styling)
* [JavaScript](https://www.browserstack.com/guide/javascript-web-development) (for interactivity)

Students can be guided through the client-side web programming with these [HTML and CSS activities.](https://www.browserstack.com/guide/build-a-website-using-html-css)

Students may investigate front-end frameworks like [React, Angular, or Vue.js](https://www.browserstack.com/guide/angular-vs-react-vs-vue) that are used to simplify development.

#### Server-side (back-end) web programming

**Activity 26:** using the space below define server-side programming and list examples of back-end languages and frameworks.

|  |
| --- |
| **Sample answer:**  Server-side web programming involves writing code that runs on the server.  This code enables the server to respond to requests from the client-side applications (such as web browsers) and generate dynamic content to be sent back to the client. It is the part of web development that deals with the server and database interactions, handles the data processing and generates the webpages that users see.  Common back-end languages and frameworks include:   * Node.js (JavaScript) with Express.js * Python with Django or Flask * Ruby with Ruby on Rails * Java with Spring Boot * PHP with Laravel |

**Activity 27:** complete the exercises in [Node JS tutorial.](https://www.w3schools.com/nodejs/default.asp)

#### Database interfacing

Interaction with databases is crucial for storing and retrieving data.

**Activity 28:** using the space below describe the 2 main types of databases used in programming for the web and provide an example of each.

|  |
| --- |
| **Sample answer:**  Two main types of databases used in programming for the web are:   1. Relational databases: relational databases organise data into tables, where each table consists of rows and columns. They follow the relational model, which means they establish relationships between tables based on keys. The Structured Query Language (SQL) is typically used to interact with relational databases (for example, MySQL).   MySQL is one of the most popular open-source relational database management systems. It's known for its reliability, scalability, and ease of use. Many web applications, including WordPress, Drupal, and Joomla, use MySQL as their database back end. MySQL is suitable for various web applications, from small-scale websites to large enterprise systems.   1. NoSQL Databases: NoSQL databases are non-relational databases that store and retrieve data in a non-tabular format. They offer flexible schema designs, scalability, and high performance for handling large volumes of unstructured or semi-structured data. NoSQL databases are commonly used in web development for real-time analytics, content management, and big data applications.   Example: MongoDB  MongoDB is a popular open-source NoSQL database that stores data in flexible JSON-like documents. It offers features like high availability, horizontal scalability, and support for complex queries. MongoDB is widely used in web development for applications requiring fast iteration and scalability, such as e-commerce platforms, social networks, and content management systems. |

**Activity 29:** complete the initial exercise in [MySQL](https://www.w3schools.com/mysql/default.asp).

**Activity 30:** complete the initial exercises in [MongoDB](https://www.w3schools.com/mongodb/index.php).

**Activity 31:** complete the IPO chart below with queries users may make of the personal interest catalogue being made for the assessment task. Start with identifying outputs (1), then inputs (2) required and finally the processes (3). In this case the process is the query.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
|  |  | 1. The result PWA users wish to find |

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 2. The tables and fields required |  | 1. The result PWA users wish to find |

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| 2. The tables and fields required | 3. The SQL query that will ordering the input into the output | 1. The result PWA users wish to find |

This IPO chart will assist students to plan and design the tables required within the database.

Each table created requires fields that consider field names and data types.

All SQL queries made in the SQL file should be documented by commenting them out progressively, so they are all recorded when developing their PWA

**Project work**

**Activity 32:** students insert data into their databases and query them to confirm the data is present (see [Appendix 1](#_Appendix_1:_Progressive)).

### Explore and explain the influence of a web browser on web development, including the use of developer (dev) tools

**Activity 33:** students investigate and create summary notes on: [30 Years of Browsers: A Quick History](https://au.pcmag.com/browsers/85834/30-years-of-browsers-a-quick-history).

**Homework**: students watch [The Dramatic History of the Web Browser (33:42)](https://youtu.be/KziWxegPrww?si=eG4MTyF1Xg7Wvbiu).

Students collaborate to create a timeline of the history of the browser.

Students rank the events into most influential milestones on web development.

Student teams explore and explain the influences of the web browser on web development [Top Web Development Tools in 2023 | BrowserStack](https://www.browserstack.com/guide/web-development-tools).

Each team provides one slide which is compiled into a study resource slide deck for this syllabus content.

Rendering engine compatibility

|  |
| --- |
| **Sample answer:**  Web developers need to ensure that their websites and web applications are compatible with various web browsers and their rendering engines (such as Blink for Google Chrome, Gecko for Mozilla Firefox, WebKit for Safari). This involves testing and debugging to ensure consistent rendering across different browsers. |

CSS and HTML standards

|  |
| --- |
| **Sample answer:**  Web browsers adhere to standards set by organizations like the World Wide Web Consortium (W3C) for HTML, CSS and JavaScript. Web developers need to write code that follows these standards to ensure cross-browser compatibility and consistent rendering. |

JavaScript execution

|  |
| --- |
| **Sample answer:**  JavaScript is executed by the browser's JavaScript engine. Different browsers may have slightly different implementations or optimizations, so developers must write JavaScript code that works efficiently across multiple browsers. |

Developer tools

|  |
| --- |
| **Sample answer:**  Modern web browsers come with built-in developer tools that provide a suite of features to aid in web development and debugging. These tools include elements inspectors for examining HTML structure, style editors for modifying CSS styles in real-time, JavaScript consoles for debugging JavaScript code, network monitors for analysing network requests and responses, and performance profilers for identifying performance bottlenecks. Developer tools also often include features for testing responsiveness, emulating different devices and simulating network conditions to ensure optimal performance across various scenarios. |

Cross-browser testing

|  |
| --- |
| **Sample answer:**  Web developers use web browsers to perform cross-browser testing to ensure that their websites and web applications work as expected across different browsers and versions. They may use tools like browser testing services or virtual machines to test their applications in various browser environments. |

Feature detection and progressive enhancement

|  |
| --- |
| **Sample answer:**  Web developers leverage the capabilities of modern web browsers to implement advanced features like CSS animations, HTML5 APIs and ECMAScript standards. Feature detection techniques are used to detect whether a particular feature is supported by the user's browser, allowing developers to provide fallbacks or alternative approaches for browsers that lack support. |

Debugging and optimisation

|  |
| --- |
| **Sample answer:**  Web browsers provide tools for debugging and optimising web applications to improve performance and user experience. Developers can use browser developer tools to identify and fix issues such as slow-loading resources, memory leaks, layout problems and JavaScript errors. |

### Investigate cascading style sheets (CSS) and their impact on the design of a web application

**Including:** **consistency of appearance, flexibility with browsers or display devices, CSS maintenance tools**

**Activity 34:** students use [W3 schools](https://www.w3schools.com/css/default.asp) to ensure consistency of appearance, and flexibility with browsers or display devices with their personal interest catalogue.

**Activity 35:** students experiment with [SASS](https://www.w3schools.com/sass/default.php) pre-processor maintenance tools to save time and reduce repetition.

**Activity 36:** students investigate [Bootstrap](https://www.w3schools.com/bootstrap/default.asp) to see how these frameworks are used in the web development industry.

### Investigate the reasons for version control and apply it when developing web application

**Activity 37:** while completing their assessment task students complete the initial tutorial exercises in W3 schools on using [Git](https://www.w3schools.com/git/default.asp).

**Activity 38**: version control, also known as source control or revision control, is a system that manages changes to documents, computer programs, large web applications and other collections of information. Use the space below and the keywords provided to explain why version control is important in each phase of software engineering.

**Keywords:** collaboration, history tracking, backup and disaster recovery, branching and experimentation, code reviews and quality assurance, deployment and continuous integration, traceability and compliance

|  |
| --- |
| **Sample answer:**  Version control allows multiple developers to work on the same project at the same time without interfering with each other's work. It enables collaboration by providing mechanisms for merging changes, resolving conflicts and tracking contributions from team members.  Version control systems maintain a history of changes made to files over time. Developers can review previous versions, compare changes between revisions and revert to earlier states if needed. This is invaluable for debugging, auditing and understanding the changes in the code.  Version control is a backup mechanism for the code. It stores copies of files on remote servers or cloud storage and protects against data loss due to hardware failure, human error or catastrophic events.  Version control systems support branching, which allows developers to create separate lines of development for experimenting with new features, fixing bugs or implementing changes without affecting the main code. Branches can be merged back into the main branch when ready.  Version control allows for code reviews by providing a centralised platform for sharing code changes and conducting peer reviews. Reviewers can provide feedback, suggest improvements and ensure code quality before merging changes into the main branch.  Version control integrates with deployment workflows and enables automated processes.  Version control systems maintain an audit trail of all changes made to the code, including who made the changes, when they were made and why. In industry, this traceability is essential for compliance with regulatory requirements, such as those required by industry standards or legal frameworks. |

### Explore the types and significance of code libraries for front-end web development

**Including: frameworks that control complex web applications, template engines, predesigned CSS classes**

Code libraries play a significant role in front-end web development by providing pre-written code, components and utilities that developers can use to streamline development, enhance functionality and improve the user experience. An example that students may investigate is [React.](https://react.dev/)

Students watch the video [What Is React And Why You Need To Know It (12:36)](https://youtu.be/1wZoGFF_oi4?si=m7wfJeLQ8NpY2jMe).

**Activity 39:** students experiment with the JavaScript library [React](https://www.w3schools.com/react/default.asp) to build user interfaces and a single-page application. They practice creating reusable UI components using React.

Front-end frameworks like React.js, Angular and Vue.js provide a structured approach to building complex web applications. They offer features such as component-based architecture and state management which help developers efficiently manage application logic and user interfaces.

**Activity 40: template engines –** students explore template engines like [Handlebars](https://handlebarsjs.com/), [Mustache](https://mustache.github.io/) and [Pug](https://pugjs.org/api/getting-started.html) used to simplify the process of generating dynamic HTML content on the client-side. These engines allow developers to create reusable templates with placeholders for dynamic data, which can be populated with actual data at runtime. Template engines facilitate code organisation, separation of concerns and maintainability by separating HTML structure from JavaScript logic, making it easier to update and modify templates without altering underlying code.

**Activity 41: predesigned CSS classes –** students explore CSS frameworks like [Bootstrap](https://bootstrapmade.com/website-templates/), [Foundation](https://get.foundation/) and [Bulma](https://bulma.io/). These provide predesigned CSS classes, components and layouts that developers can leverage to create responsive and visually appealing web interfaces. These frameworks offer grids, typography, forms, buttons, navigation bars and other UI components out of the box, allowing developers to rapidly prototype and build consistent user interfaces. By using predesigned CSS classes, developers can save time, ensure cross-browser compatibility and maintain a cohesive visual design across their web applications.

**Activity 42:** using the keywords below, discuss the significance of code libraries for front-end web development. Answer in the space below. to provide your answer.

**Keywords:** efficiency, consistency, scalability, maintainability, accessibility

|  |
| --- |
| **Sample answer:**  Code libraries enable developers to reuse existing code components and utilities. This reduces development time and effort. Frameworks and CSS libraries provide predefined conventions and styles that ensures consistency in design and behaviour across different parts of the web application. Libraries and frameworks offer scalable solutions for building complex web applications, they provide tools and patterns for managing application architecture, state and data flow. Template engines and modular CSS frameworks promote code organisation; they separate concerns and maintenance across teams and make it easier to manage and update code over time. Many front-end libraries and frameworks prioritise accessibility by providing built-in features and best practices for creating inclusive user interfaces that are usable by all users, including those with disabilities. |

**Teacher note:** the code libraries referenced above were available and prevalent at the time of writing. Teachers are advised to check in on latest web developments and update resources accordingly.

### Explain the use and development of open-source software in relation to web development

**Activity 43:** students visit and read [A Brief History of Open Source](https://www.freecodecamp.org/news/brief-history-of-open-source/).

Using research and references from the history of open source, explain the use and development of open-source software in relation to web development. Write the explanation in the space provided below.

|  |
| --- |
| **Sample answer:**  Open-source software plays a pivotal role in web development by democratising access to technology, fostering collaboration and innovation, promoting transparency and security, and driving the evolution of the web development ecosystem. By embracing open-source principles and contributing to open-source projects, developers can harness the collective power of the community to build better, more sustainable and more impactful web applications for the benefit of all. |

**Activity 44:** class debate.

Who made a more important contribution to computing: Bill Gates or Linus Torvalds?

### Investigate methods to support and manage the load times of webpages/applications

**Activity 45:** Why is it important to manage load times of webpages/applications? Answer in the space below.

|  |
| --- |
| **Sample answer:**  Managing load times for webpages and applications is essential for creating a positive user experience. Otherwise, users may visit the page and leave straight away. The percentage of visitors to a website who navigate away from the site after viewing only one page, without interacting with any other pages or elements on the site is called the bounce rate. Improved load times reduces bounce rates and improving search engine rankings. |

**Activity 46:** What are some of the methods to manage load times of webpages/applications? Answer in the space below.

|  |
| --- |
| **Sample answer:**  Student responses might vary, but could include:   * Compress images to reduce file sizes without significantly compromising quality. * Use responsive images and defer the loading of off-screen images until they are needed. * Reduce the number of HTTP requests by combining multiple CSS and JavaScript files into a single file each. * Allow browsers to store cached copies of resources locally, reducing the need to re-download them on subsequent visits. * Remove unnecessary whitespace, comments, and formatting, reducing file sizes and improving load times. * Use performance monitoring tools to identify and address server-side performance bottlenecks. * Compression on the server to compress text-based resources such as HTML, CSS, and JavaScript files before transmitting them to clients. |

### Research, experiment with and evaluate the prevalence and use of web content management systems (CMS)

**Activity 47:** What is a web content management system (CMS)? Answer in the space provided.

|  |
| --- |
| **Sample answer:**  Web content management systems (CMS) are software platforms that allow users to create, manage and publish digital content on the web without requiring extensive technical knowledge.  They are widely used across various industries and sectors to facilitate website development and maintenance. |

**Activity 48:** research and evaluate the prevalence and use of web content management systems (CMS). Identify the most popular CMS platforms and their market share. Answer in the space provided.

|  |
| --- |
| **Sample answer:**   1. WordPress: holds a market share of 62.7%. WordPress [powers a significant portion of websites globally and offers a vast library of themes and plugins](https://www.tooltester.com/en/blog/cms-market-share/). 2. Shopify: holds a market share of 6.4%. It is known for e-commerce, but also serves as a CMS. 3. [Wix: holds a market share of 3.9%. It is known for its user-friendly interface and drag-and-drop website building capabilities](https://www.mobiloud.com/blog/cms-market-share). 4. [Squarespace: holds a market share of 3.0%. It is popular for its elegant templates and ease of use](https://www.mobiloud.com/blog/cms-market-share). 5. [Joomla: holds a 2.4% market share. It has seen declining growth compared to WordPress](https://www.tooltester.com/en/blog/cms-market-share/). |

**Teacher note:** the above were the statistics on market share at the time of writing.

**Activity 49:** experiment with [WordPress](https://wordpress.com/) and take notes and screenshots of the products, features and resources that could be of use for your assessment task.

### Assess the contribution of back-end web development to the success of a web application

Using the [flipped classroom approach](https://www.teaching.unsw.edu.au/flipped-classroom), students watch [Backend web development – a complete overview (12:57)](https://youtu.be/XBu54nfzxAQ?si=10MpOwdB_Q_VzaPc) for homework.

**Teacher note**: This comprehensive description of back-end web development goes beyond the requirement of the syllabus. It could be used as an extension or to provide students with insight into the processes they may use in developing a major project based around programming for the web.

**Front end vs back end**

**Activity 50:** complete the table according to the video description.

|  |  |
| --- | --- |
| Front end | Back end |
| ‘The front end is all the visual stuff you see on the webpage.’ | ‘The back end is what saves and manages your data.’  If you are on amazon.com, the back end would store your order history and your profile. It would load search results and much more. |

**Activity 51:** using the online shopping scenario from the video, describe what a server is. Answer in the space provided.

|  |
| --- |
| **Sample answer:**  To shop on amazon.com, a customer clicks to place an order.  The customer’s computer is connected to the internet.  The computer sends a message across the internet to Amazon’s computer that's also connected to the internet. In this scenario, the computer that is sending the message is called the ‘client’, and the computer that is receiving the message is called the ‘server’. |

**Activity 52:** explain why a back-end programming language is needed. Answer in the space provided.

|  |
| --- |
| **Sample answer:**  Computers can't receive messages from the internet by default.  They must be programmed to be able to receive messages. To do that, it requires a back-end programming language. Almost every programming language has a feature that turns a computer into a server and allows it to receive messages. Popular back-end programming languages are JavaScript, sometimes called node js, Python, Ruby, and Java. |

**Activity 53:** describe what a back-end framework does and provide examples. Answer in the space provided.

|  |
| --- |
| **Sample answer:**  Using a back-end programming language by itself is difficult and requires a large amount of code. There are 2 tools to help with this. A back-end framework and a package manager. A back-end framework helps create a server much easier and with a lot less code. Each back-end programming language has a few different frameworks to choose from, but the most popular ones are Express JS for JavaScript, Python, Django, Rubion Rails and Java Spring. |

**Activity 54:** explain what a package manager does and provide examples. Answer in the space provided.

|  |
| --- |
| **Sample answer:**  The back end has code that other people have written called packages to do common tasks like doing calculations, talking to a database and setting up user login and authentication. A lot of packages are used in the back end, and to install and manage all these packages, a package manager is used. Each language has its own package manager. The most popular ones are NPM for JavaScript, PIP for Python, bundler for Ruby and Maven for Java. These are all the technologies needed to create a back-end server. |

**Activity 55**: What is the role of the database in the Amazon scenario? Answer in the space provided.

|  |
| --- |
| **Sample answer:**  When shopping online, there needs to be a space to save the data for the website.  In the Amazon example, data includes user data, like the login information, order history, as well as data for all the products that are being sold on Amazon, such as the descriptions, ratings and reviews. A database is used to save this data. A database helps store and manage data. It is software that usually runs on a different computer. Some setup is required so that the back end can communicate with the database. The most popular databases are MySQL, Postgres and MongoDB. |

**Activity 56** explain the request response cycle in the Amazon scenario? Answer in the space provided.

|  |
| --- |
| **Sample answer:**  When the customer places an order in the front end, the front end sends a message containing the order to the back end. The back end then saves the order to a database and sends back a message to the front end confirming that the order was created. The message that the front end sends to the back end is known as a request, and the message that the back end sends back is known as a response. This is called a request response cycle, and this is generally how web applications work. |

**Activity 57:** explain API’s (application programming interface) using the Amazon scenario. Answer in the space provided.

|  |
| --- |
| **Sample answer:**  A simplified example of a request to create an Amazon order has the following inside:  The items ordered, the quantities and other information about the Amazon order.  At the top is shown the type of the request, the domain name and the URL path. This describes where this request is going and what type of request it is. Since Amazon (the company) bought the domain name amazon.com, it is configured it so that any request going to amazon.com will be redirected to the server in their office building. The request goes to amazon.com. The type and the URL path identify what kind of request this is.  An example is: POST https://amazon.com/orders  In the back end, the programming language and back-end framework define what types of requests are allowed and how to handle these requests.  POST https://amazon.com/orders is allowed and whenever a post/order is requested, an order is created using the programming language and saved to the database.  A get/order request will retrieve the order history from the database and send it back as a response.  A delete/order request will cancel the order.  These different types of requests that the back end allows is called an API.  The API is one of the most important concepts in back-end programming.  If a request is sent that is not allowed by the API, the back end will respond with an error. |

**Activity 58:** Unplugged activity

Students form small teams to role-play the processes involved in [Alice shopping online – Mapping out a request](https://www.codecademy.com/article/back-end-architecture#:~:text=Mapping%20out%20a%20request).

**Activity 59:** extension activity – students complete notetaking for the remainder of the video [Backend web development – a complete overview (12:57)](https://youtu.be/XBu54nfzxAQ?si=10MpOwdB_Q_VzaPc) and research the following:

* REST API
* Cloud computing and IaaS (Infrastructure as a Service)
* 7VMs and Load Balancers
* PaaS (Platform as a Service)
* Microservices
* SaaS (Software as a Service)
* additional technologies.

**Teacher note:** software engineering teachers could approach their TSO’s or computer co-ordinators for an old laptop to recycle into a web server. A desktop or microcontroller (for example, raspberry pi) could also be used. Assuming an old laptop is used and running Windows, XAMPP could be installed which would provide a webserver, PHP and SQL. Node.js can then be installed as well.

The learning experience by establishing a web server could inform students considering extending on the Programming for the web focus area for their major projects. They will deepen their knowledge of the ‘plumbing’ and coding required at both front and back ends of the web.

Please refer to Appendix 2 – web server guide.

### Observe and describe the back-end process used to manage a web request

**Including: role of webserver software, web framework, objects, libraries, databases**

Observing and describing the back-end process used to manage a web request involves understanding the roles of various components such as web server software, web framework objects, libraries and databases.

**Activity 60:** research and provide an overview of the back-end process to manage a web request including the role of the web server, web framework, objects, libraries and databases. Answer in the space provided.

|  |
| --- |
| **Sample answer:**  Process overview:  The webserver software receives the incoming request and forwards it to the appropriate endpoint in the web framework.  The web framework processes the request, routes it to the corresponding view or controller function, and interacts with objects, libraries and databases to handle the request logic.  Objects and libraries are used within the back-end code to perform specific tasks, such as data processing, validation or external integrations.  Databases are queried and updated to retrieve or store data related to the request, allowing the back end to generate a response based on the data retrieved and processed. |

**Activity 61:** web server software is responsible for receiving incoming HTTP requests from clients (browsers or applications) and forwarding them to the appropriate components for processing. Study the following code.

// Import required modules

const express = require('express');

// Create an instance of Express.js

const app = express();

// Define a route to handle incoming HTTP GET requests

app.get('/', (req, res) => {

    // Send a plain text response to the client

    res.send('Hello, world!');

});

// Start the server and listen on port 3000

app.listen(3000, () => {

    console.log('Server is running on http://localhost:3000');

});

In the space below, describe what this code does.

|  |
| --- |
| **Sample answer:**  The **express** module is imported and provides functionalities for building web servers and handling HTTP requests/responses. An instance of Express.js is created using **express()**.The **app.get()** method defines the route specifying that it should handle HTTP GET requests to the root URL (**'/'**). When such a request is received, the provided callback function is executed. Inside the callback function, the **res.send()** method is used to send a plain text response (**'Hello, world!'**) back to the client. The server started by calling the **app.listen()** method, specifying the port number (in this case, **3000**) on which the server should listen for incoming connections. This simple web server will respond with "Hello, world!" to any HTTP GET requests it receives on the root URL (**http://localhost:3000/**). |

### Develop a web application using an appropriate scripting language with shell scripts to make files and directories, and searching for text in a text file

When developing a web application that involves creating files and directories, as well as searching for text in text files using shell scripts, a suitable scripting language to use is Bash (Bourne Again Shell).

Bash is a widely used shell scripting language in Unix-based systems like Linux and macOS, offering file manipulation, directory operations and text processing.

Bash provides a rich set of commands and utilities for **creating files and directories, moving, copying, and deleting files, as well as setting permissions and ownership**. **Commands like mkdir for creating directories, touch for creating files, mv for moving files, cp for copying files, and rm for deleting files are commonly used in Bash scripts**. **Bash offers powerful text processing capabilities, allowing you to search for specific text patterns in files, extract data, and manipulate text content. Commands like grep for searching text in files based on patterns, sed for text substitution and manipulation, and awk for text processing and reporting are commonly used for text processing tasks.**

Bash enables seamless integration with system commands and utilities, enabling developers to combine shell commands with file operations and text processing within the script. Developers can execute system commands, capture their output and process the data using Bash scripting.

Bash scripts are portable across Unix-based systems, making them a versatile choice for automating tasks and system administration across different environments.

**Activity 62:** use Git Bash to experiment with scripting.

Refer to Initialize Node.js application in the PWA in Appendix 1.

Git Bash is a command-line terminal application that provides a Unix-like environment on Windows systems.

**Activity 63:** **homework** – watch [Bash Scripting Tutorial for Beginners (47:56)](https://www.youtube.com/watch?v=tK9Oc6AEnR4) for more #BASH knowledge.

### Apply a web-based database and construct script that executes SQL

**Including: selecting fields, incorporating ‘group by’, common SQL queries, constraints using WHERE keyword, table joins**

Refer to ‘**Build and test query your database’** from the Appendix 1 – progressive web application.

**Activity 64:** students will create the database based on the catalogue required for their PWA project.

Teacher note: this may have already been completed by students. They should be referred to this syllabus content and attempt each of the SQL statements.

Students can study the sample code for a music catalogue before completing their own:

**# Create a new SQLite database file**

sqlite3 music\_catalogue.db

# Create a 'tracks' table in the database

sqlite> CREATE TABLE tracks (

...> id INTEGER PRIMARY KEY,

...> title TEXT,

...> artist TEXT,

...> genre TEXT,

...> release\_date DATE

...> );

**# Insert sample data into the 'tracks' table**

sqlite> INSERT INTO tracks (title, artist, genre, release\_date) VALUES ('Song 1', 'Artist A', 'Pop', '2021-01-01');

sqlite> INSERT INTO tracks (title, artist, genre, release\_date) VALUES ('Song 2', 'Artist B', 'Rock', '2020-06-15');

# Select fields from the 'tracks' table

sqlite> SELECT title, artist, genre, release\_date FROM tracks;

**# Group tracks by genre**

sqlite> SELECT genre, COUNT(\*) as track\_count FROM tracks GROUP BY genre;

# Select tracks by a specific artist

sqlite> SELECT title, genre FROM tracks WHERE artist = 'Artist A';

# Select tracks released after a specific date

sqlite> SELECT title, genre FROM tracks WHERE release\_date > '2021-01-01';

**# Table join between 'tracks' and 'artists' tables** (assuming 'artists' table exists)

sqlite> SELECT tracks.title, artists.name FROM tracks JOIN artists ON tracks.artist = artists.id;

### Compare Object-Relational Mapping (ORM) to SQL

**Activity 65:** students watch [Introduction to Object-Relational Mapping (ORM) (11:35)](https://youtu.be/dHQ-I7kr_SY) and read the text below.

ORM (Object-Relational Mapping) and SQL (Structured Query Language) are both methods used in software development for working with databases, but they serve different purposes and have distinct advantages and limitations.

ORM simplifies database interaction and enhances productivity by abstracting database operations into object-oriented constructs, while SQL provides fine-grained control over database operations and performance optimisation.

The choice between ORM and SQL depends on factors such as project requirements, developer preferences, performance considerations and familiarity with database technologies.

Consider an example of managing a music catalogue using both Object-Relational Mapping (ORM) and SQL to illustrate their differences:

#### Using ORM (Object-Relational Mapping)

In an ORM framework like Django ORM in Python, a model class can be defined for a music track in the catalogue. An example of a simplified model class in Django:

from django.db import models

class MusicTrack(models.Model):

title = models.CharField(max\_length=100)

artist = models.CharField(max\_length=50)

genre = models.CharField(max\_length=50)

release\_date = models.DateField()

With this model class, Django ORM automatically creates a corresponding database table in the relational database, based on the defined fields.

Developers can interact with the database using Python objects without directly writing SQL queries.

For example, to retrieve all tracks in the 'Rock' genre:

rock\_tracks = MusicTrack.objects.filter(genre='Rock')

#### Using SQL

When using SQL directly, developers write queries to interact with the database tables. Here is an example of a SQL query to retrieve tracks in the 'Rock' genre from a 'music\_tracks' table:

SELECT \* FROM music\_tracks WHERE genre = 'Rock';

Developers have full control over the SQL queries, allowing them to optimise queries for performance, join tables and perform complex operations. SQL provides fine-grained control but requires a deeper understanding of the database schema and query syntax.

**Activity 66:** explain the differences between ORM and SQL using the following categories: Abstraction, Flexibility, Performance, Learning curve. Answer in the space below.

|  |
| --- |
| **Sample answer:**  **Abstraction**: ORM abstracts database operations through object-oriented models, while SQL requires writing detailed queries.  **Flexibility**: ORM provides flexibility by simplifying database interactions with objects, while SQL offers fine-grained control over queries.  **Performance**: ORM may introduce overhead due to the abstraction layer, while SQL allows developers to optimise queries for performance directly.  **Learning curve**: ORM frameworks like Django ORM have a learning curve, while SQL syntax is more straightforward and widely used.  In the music catalogue example, using ORM simplifies database interactions by mapping objects to database tables, while using SQL directly offers greater control over queries and optimisation. The choice between ORM and SQL depends on the project requirements, development team expertise and the desired level of abstraction versus control in managing a music catalogue or any database-driven application. |

### Describe how collaborative work practices between front-end and back-end developers improve the development of a web solution



**Activity 67:** [Think-Pair-Share](https://app.education.nsw.gov.au/digital-learning-selector/LearningActivity/Card/645?clearCache=befbfa2e-bd65-40ce-da3f-b12bf11851b5) a description of each of the collaborative work practices shown on the tiles above.

|  |
| --- |
| **Sample answer:**  **Shared understanding:** front-end and back-end developers have different areas of expertise. By working collaboratively, they can develop a shared understanding of the project requirements, design considerations, and technical constraints.  **Cross-functional collaboration:** collaborative work practises between front-end and back-end developers allows team members to leverage each other's strengths, share knowledge, and collectively solve complex problems that require expertise from both areas.  **Continuous communication:** regular discussions, stand-up meetings, and feedback sessions help ensure that both teams are aligned on project milestones, progress, and any potential issues that may arise.  **Iterative development:** collaborative work practises facilitate iterative development cycles where front-end and back-end developers work together to build, test, and refine features incrementally. An iterative approach allows for early feedback, rapid iteration, and continuous improvement based on user input and changing requirements.  **Code consistency and integration:** front-end and back-end developers can ensure code consistency and seamless integration between the front-end user interface and the back-end server-side logic. This reduces the risk of miscommunication, conflicts, or discrepancies in the codebase.  **Problem-solving and troubleshooting**: front-end and back-end developers collaborate on problem-solving and troubleshooting tasks effectively. By sharing insights, expertise, and perspectives, both teams can diagnose issues, identify root causes, and implement solutions in a coordinated manner.  **Quality assurance and testing:** front-end and back-end developers collaborate to ensure comprehensive test coverage, validate functionality across the entire application stack, and address any issues that may impact user experience or system performance.  **Cross-training and skill development:** collaboration between front-end and back-end developers provides opportunities for cross-training and skill development. By sharing knowledge, best practises and learning from each other's expertise, team members can enhance their skills, broaden their understanding of technology stacks and improve overall proficiency. |

### Design, develop and implement a progressive web app (PWA)

**Including: the application of design and user interface (UI) and user experience (UX) principles of font, colour, audio, video and navigation and a UI that considers accessibility and inclusivity**

**Activity 68:** research each of the UX and UI principles.

Develop a scorecard to self and peer assess the PWAs by providing numerical weightings to the priorities.

|  |  |  |  |
| --- | --- | --- | --- |
| UI principles | Priority | UX principles | Priority |
| Responsive design | High | Fast loading times | High |
| Intuitive navigation | High | Clear calls to action | High |
| Consistent design language | High | Offline accessibility | High |
| App-like experience | Medium | Accessibility | High |
| Clear calls to action | Medium | Feedback and error handling | Medium |
| Font readability | Medium | User testing and iteration | Medium |
| Colour contrast | Medium |  |  |
| Audio integration | Low |  |  |
| Video integration | Low |  |  |
| Navigation design | Low |  |  |

# Appendix 1 – progressive web application

## Introduction to PWA

‘A [progressive web app (PWA)](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps?ref=arctype.com) is an app that's built using web platform technologies, but that provides a user experience like that of a platform-specific app.

Like a website, a PWA can run on multiple platforms and devices from a single codebase. Like a platform-specific app, it can be installed on the device, can operate while offline and in the background, and can integrate with the device and with other installed apps.’ (Mozilla n.d.)

### Technical features of PWAs

Because PWAs are websites, they have the same basic features as any other website: at least one HTML page, which very probably loads some CSS and JavaScript. Like a normal website, the JavaScript loaded by the page has a global Window object and can access all the Web APIs that are available through that object.

#### Beyond that, a PWA has some additional features:

* A web app manifest file, which, at a minimum, provides information that the browser needs to install the PWA, such as the app name and icon.
* A service worker, which, at a minimum, provides a basic offline experience.

Read more about how a PWA is different to a traditional website or a platform-specific website [here](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/What_is_a_progressive_web_app).

### Setup Web Developer Environment and Folder Structures

#### Introduce the task

Download the source code (or have your own example): <https://github.com/TempeHS/Programming_For_The_Web_Task_Source>

The development process for this example PWA has been specifically designed to demonstrate specific content points from the [Software Engineering 11–12 Syllabus](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/overview):

* Model elements that form a web development system
* Investigate cascading style sheets (CSS) and its impact on the design of a web application
* Investigate methods to support and manage the load times of webpages/applications
* Observe and describe the back-end process used to manage a web request
* Develop a web application using an appropriate scripting language with shell scripts to make files and directories, and searching for text in a text file
* Apply a web-based database and construct script that executes SQL
* Design, develop and implement a progressive web app (PWA).

## Set up your development environment

The following applications are required, as a minimum, to complete this task.

|  |  |
| --- | --- |
| Check | Step |
|  | * Install Visual Studio Code |
|  | * Install Node.Js * <https://nodejs.org/en/download> |
|  | * Install VSCode Extensions * <https://marketplace.visualstudio.com/items?itemName=alexcvzz.vscode-sqlite> * <https://marketplace.visualstudio.com/items?itemName=McCarter.start-git-bash> * <https://marketplace.visualstudio.com/items?itemName=miramac.vscode-exec-node> * Optional and recommended extensions * <https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css> * <https://marketplace.visualstudio.com/items?itemName=oderwat.indent-rainbow> * <https://marketplace.visualstudio.com/items?itemName=ms-vscode.js-debug-nightly> * <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer> * <https://marketplace.visualstudio.com/items?itemName=yy0931.vscode-sqlite3-editor> * Window users only * <https://marketplace.visualstudio.com/items?itemName=medo64.render-crlf> |

If you are new to VSCode or don’t know how to install extensions use [this guide](https://code.visualstudio.com/docs/introvideos/basics).

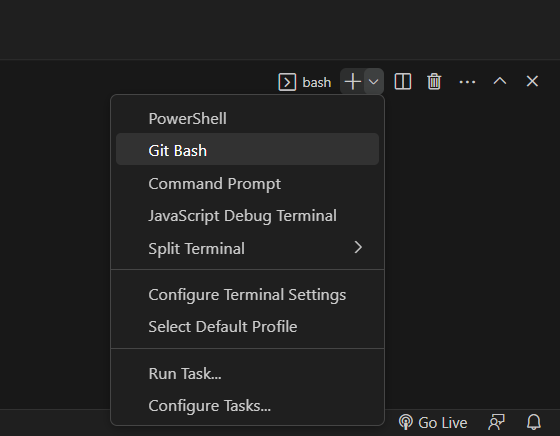
**Note:** VSCode and Node.JS are available in the eT4L software catalogue.

## Establish a working directory

1. Start by selecting an appropriate folder location that is either version controlled or backed up to the cloud.
2. Open VSCode Ctrl + K Ctrl + O
3. Navigate to and open your working folder.

## Initialise Node.js application

To get started, we'll create our project folder with the #bash command below. Make sure you open a new terminal Ctrl + ` and choose **Git Bash** from the menu option in the top right of the terminal shell.



Start by making a folder with the following #BASH commands.

mkdir myPWA && cd myPWA

Then, we'll initialise a Node.js application with the commands below:

npm init -y

The above command creates and configures a package.json file for the application.

## Set up files and folders for your project

|  |  |
| --- | --- |
| Practical activities | Theory |
| Create a directory for development files and documentation  mkdir .workingDocuments  Create a license file  touch LICENSE  Edit the LICENSE document and paste the text from [The GNU General Public License (version 3)](https://www.gnu.org/licenses/gpl-3.0.txt) into it.  Create a directory for your database and SQL files  mkdir .database  Create a public directory for your front-end files with the #bash command below:  mkdir public && cd public  Now using #BASH commands and scripts setup a template folder structure for a PWA  touch files.txt  touch folders.txt  Inside the file, list the files you want to routinely generate:   1. style.css 2. index.html 3. index.js 4. app.js 5. serviceWorker.js   Then create a file called folders.txt and in the file list the folders you want to routinely generate:   1. css 2. icons 3. images 4. js   Write a simple bash script to read the folders.txt document and create a directory for each line.  while read -r line; do  echo $line  mkair -p $line  done < folders.txt  Write a simple bash script to read the files.txt document and create a file for each line.  while read -r line; do  echo $line  touch $line  done < files.txt | Watch [Bash Scripting Tutorial for Beginners (47:56)](https://www.youtube.com/watch?v=tK9Oc6AEnR4) for more #BASH knowledge.  See the [Bash cheat sheet: Top 25 commands and creating custom commands](https://www.educative.io/blog/bash-shell-command-cheat-sheet) for common #BASH commands.  **Notes:**   * The dot .file instructs the webserver to not serve the folder or contained files securing the data from being accessed via http adding an additional layer of secure architecture. * The Blank line is required to give the last line a line ending and allow the script below to read all lines. * If you are using a DOS (windows) environment with a bash emulator in VSCode the line ending will be incorrect you will need to use an online converter to replace the DOS line endings with Unix line endings such as [*https://app.execeratics.com/LFandCRLFonline/?l=en*](https://app.execeratics.com/LFandCRLFonline/?l=en)you can see your line endings using the optional extension listed in the setting up your environment section. |

## Build and test query your database

|  |  |
| --- | --- |
| Practical activities | Theory |
| Install SQLite3 via NPM at the #BASH command.  npm install --save sqlite3  Go to the .database folder and create your database files.  touch datasource.db  touch myQuery.sql  Edit the myQuery.sql file and write the query to create the table.  CREATE TABLE extension(extID INTEGER NOT NULL PRIMARY KEY,name TEXT NOT NULL, hyperlink TEXT NOT NULL,about TEXT NOT NULL,image TEXT NOT NULL,language TEXT NOT NULL);  Run the query by right click over the query then choose “Run Selected Query” and choose datasource.db when ask this will associate all queries in myQuery.sql with the database datasource.db.  Write a template for inserting values.  -- INSERT INTO extension(extID,name,hyperlink,about,image,language) VALUES (X,"","","","",""); -- Insert template  Then copy and paste it filling in the insert values to manually add content to your database.  INSERT INTO extension(extID,name,hyperlink,about,image,language) VALUES (1,"Live Server","https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer","Launch a development local Server with live reload feature for static & dynamic pages","https://ritwickdey.gallerycdn.vsassets.io/extensions/ritwickdey/liveserver/5.7.9/1661914858952/Microsoft.VisualStudio.Services.Icons.Default","HTML CSS JS"); # V0.9 Ben Jones 15/8/2023  Once you have built your Database test it with some different QUERYs for example:  SELECT \* FROM extension;  SELECT \* FROM extension WHERE language LIKE '#BASH'; | Watch [How to Run SQLITE in Visual Studio Code (8:59)](https://www.youtube.com/watch?v=JrAiefGNUq8) to see how SQL Queries and SQLite3 works in VSCode. **Note:** after each SQL query, students should comment out -- the query so there is a running log of SQL queries. |

## Set up your Express webserver

|  |  |
| --- | --- |
| Practical activities | Theory |
| In a BASH terminal, set up a node.js file that will hold all the back-end js code, this file should be located in your application directory root.  cd .. && touch index.js  Use BASH to install the Express webserver via npm.  npm install express | **Note:**  cd.. is assuming you are working in the public folder  Read the Node.js Express documentation [here](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/development_environment). |
| Add the following JS to your index.js file:  const express = require("express");  const path = require("path");  const app = express();  app.use(express.static(path.join(\_\_dirname, "public")));  app.get("/", function (req, res) {  res.sendFile(path.join(\_\_dirname, "public/index.html"));  });  app.listen(8000, () => console.log("Server is running on Port 8000, visit <http://localhost:8000/> or <http://127.0.0.1:8000> to access your website") ); | In this script, we import express to create our server and the path module. We configured our app to render our static files using the express.static method, which takes the path to the static folder (public), we created the root route of our application and rendered the index.html file. Then we configured the app to listen to port 8000. |

Extract your SQL in the back end to a JSON in the front end.

|  |  |
| --- | --- |
| Task | Script |
| Core HTML | <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8" />    <meta name="viewport" content="width=device-width, initial-scale=1.0" />    <meta http-equiv="X-UA-Compatible" content="ie=edge" />    <meta http-equiv="Content-Security-Policy" content="script-src 'self';">    <link rel="stylesheet" href="css/style.css" />    <title>Your Tab Title Here</title>    <link rel="manifest" href="manifest.json" />    <link rel="icon" type="image/x-icon" href="images\favicon.png">    <meta name="theme-color" content="#14E6DD" />  </head>  <body>    <main>      </main>    <script type="module" src="js/app.js"></script>  </body>  </html> |
|  | Use the JSONLint editor to format and validate your JSON  https://jsonlint.com/ |

## Render your front end

1. Design a simple square logo (1080px ×1080px) using Canva or Photoshop & a simplified favicon icon 1080px ×1080px.
2. Export both files as a PNG.
3. Save design and original export files to the working files directory.
4. Optimize your logo using <https://tinypng.com/> and save the optimised versions to public/images.

|  |  |
| --- | --- |
| Task | Script |
| Core HTML | <!DOCTYPE html>  <html lang="en">  <head>    <meta charset="UTF-8" />    <meta name="viewport" content="width=device-width, initial-scale=1.0" />    <meta http-equiv="X-UA-Compatible" content="ie=edge" />    <meta http-equiv="Content-Security-Policy" content="script-src 'self';">    <link rel="stylesheet" href="css/style.css" />    <title>Your Tab Title Here</title>    <link rel="manifest" href="manifest.json" />    <link rel="icon" type="image/x-icon" href="images\favicon.png">    <meta name="theme-color" content="#14E6DD" />  </head>  <body>    <main>      </main>    <script type="module" src="js/app.js"></script>  </body>    </html> |
| Style the core | @import url("https://fonts.googleapis.com/css?family=Nunito:400,700&display=swap");  \* {  margin: 0;  padding: 0;  box-sizing: border-box;  }  body {  background: #fdfdfd;  font-family: "Nunito", sans-serif;  font-size: 1rem;  }  main {  max-width: 900px;  margin: auto;  padding: 0.5rem;  text-align: center;  } |
| Add Navigation | <main>        <nav>          <img src="images\logo.png"/>          <h1>VSCode Extensions</h1>          <ul class="topnav">            <li><a href="#">Home</a></li>            <li><a href="about.html">About</a></li>            <li><a href="suggest.html">Suggest</a></li>          </ul>        </nav>      </main> |
| Style navigation | nav {      display: flex;      justify-content: space-between;      align-items: center;  }  nav img {      height:100px;  }  nav ul {      list-style: none;      display: flex;  }  nav li {      margin-right: 1rem;    }  nav ul li a {      text-decoration-line:none;      text-transform: uppercase;      color: #393b45;    }  nav ul li a:hover {      color: #14E6DD;  }  nav h1 {      color: #106D69;      margin-bottom: 0.5rem;  } |
| Add the div container that will be scripted into below the navigation | <div class="container">          </div> |
| Style the grid | .container {  display: grid;  grid-template-columns: repeat(auto-fit, minmax(15rem, 1fr));  grid-gap: 1rem;  justify-content: center;  align-items: center;  margin: auto;  padding: 1rem 0;  }  .card {  display: flex;  align-items: center;  flex-direction: column;  width: 15rem auto;  height: 15rem;  background: #fff;  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.19), 0 6px 6px rgba(0, 0, 0, 0.23);  border-radius: 10px;  margin: auto;  overflow: hidden;  }  .card--avatar {  width: 100%;  height: 10rem;  object-fit: cover;  }  .card--title {  color: #222;  font-weight: 700;  text-transform: capitalize;  font-size: 1.1rem;  margin-top: 0.5rem;  }  .card--link {  text-decoration: none;  background: #db4938;  color: #fff;  padding: 0.3rem 1rem;  border-radius: 20px;  } |

**Note:** read more about Content Security Policy and the head meta element <meta http-equiv="Content-Security-Policy" content="script-src 'self';"> here: [Content Security Policy (CSP)](https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP).

## Running an Express server via Node.js from VSCode

#### A step-by-step guide

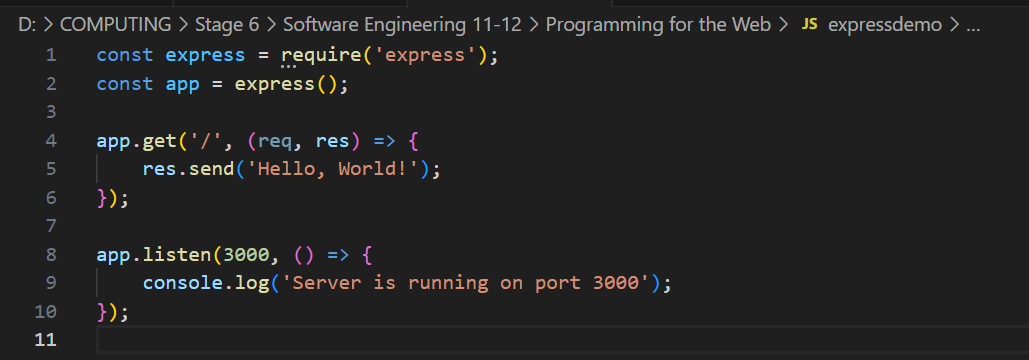
1. Open your project in VSCode:
2. Launch Visual Studio Code (VSCode).
3. Open your project folder by selecting **File** > **Open Folder** and navigating to the folder containing your Express.js project.
4. Install Node.js and npm:
5. Ensure you have Node.js and npm (Node Package Manager) installed on your computer. You can download and install them from the official [Node.js website](https://nodejs.org/).
6. After installation, you can verify that Node.js and npm are installed correctly by opening a terminal (integrated terminal in VSCode or any other terminal) and running the following commands:

**node -v**

**npm -v**

1. Install required dependencies:
2. If you haven't already done so, navigate to your project directory in the terminal and run npm install to install the necessary dependencies listed in your project's package.json file.
3. Create your Express server:
4. Write your Express server code in a JavaScript file (for example, server.js) within your project directory. If you don't have one yet, you can create a new file and write your server code there.

Here's a simple example of an Express server code:



1. Run your Express server:

Open a terminal in VSCode by selecting **View** > **Terminal** or by pressing Ctrl + ~ (backtick).

Navigate to your project directory using the cd command.

Once you're in the project directory, run the following command to start your Express server:

**node server.js**

Replace server.js with the filename where your server code is located if it's different.

1. Access your Express server:

Once your server is running, you can access it by opening a web browser and navigating to http://localhost:3000

(assuming you're using port 3000 in your express server code).

When changes are made to the server code, save the file, and restart the server to see the changes in action.

# Appendix 2 – web server guide

**Teacher note:** this unit and accompanying assessment task requires teachers to run an express server via NodeJS from VSCode.

This express server ensures delivery of the syllabus content in [Programming for the web](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/content/n12/fa6aab137e) is done with an authentic, engaging and project based approach.

Software engineering teachers **may** approach their TSO’s or computer co-ordinators for an old laptop to recycle into a web server.

A desktop or raspberry pi could also be used.

This web server would provide a learning experience that informs students considering extending on the Programming for the web focus area for their major project.

They will deepen their knowledge of the ‘plumbing’ and coding required at both front and back ends of the web.

## Web server guide

### XAMPP

* Is a toolbox of applications put together in one package
* Makes it easy to set up a local web development environment on a computer
* Is used to build and test websites before publishing them online
* Is ideal for beginners because it provides everything needed to get started with web development without having to install each component separately
* Is free and open-source, so can be used without any cost

This guide assumes that the recycled laptop is running Windows.

1. Download XAMPP
2. Open your web browser and visit the [official XAMPP website](https://www.apachefriends.org/index.html).
3. Select **XAMPP for Windows** to download the installer.
4. Install XAMPP
5. Select the downloaded file (usually named something like xampp-win32-7.2.4-0-VC15-installer).
6. Select **Yes** when prompted.
7. Select **Next** in the setup window.
8. Review the list of XAMPP components and uncheck any attributes you don’t want to install.
9. Select an installation location (avoid installing in your hard drive’s folder).
10. Select **Next** and uncheck the **Learn more about Bitnami** box.
11. Select **Next** again to start the installation.
12. Select **Finish** when prompted.
13. Configure XAMPP
14. Select your preferred language (English or German).
15. Select **Save** to open the XAMPP control panel.
16. Start XAMPP
17. To open the XAMPP control panel later, find the installation folder, right-click the orange-and-white xampp-control icon, choose **Run as administrator** and select **Yes**.

XAMPP is ready to use for local web development.

To install node.js on a Xampp localhost, follow the guidelines offered in [node.js – How to install nodejs on Xampp localhost – Stack Overflow](https://stackoverflow.com/questions/18136304/how-to-install-nodejs-on-xampp-localhost).

# References

This resource contains NSW Curriculum and syllabus content. The NSW Curriculum is developed by the NSW Education Standards Authority. This content is prepared by NESA for and on behalf of the Crown in right of the State of New South Wales. The material is protected by Crown copyright.

Please refer to the NESA Copyright Disclaimer for more information [https://educationstandards.nsw.edu.au/wps/portal/nesa/mini-footer/copyright](https://aus01.safelinks.protection.outlook.com/?url=https%3A%2F%2Feducationstandards.nsw.edu.au%2Fwps%2Fportal%2Fnesa%2Fmini-footer%2Fcopyright&data=05%7C01%7CCaitlin.Pace1%40det.nsw.edu.au%7C9c2c1a9f59c94d2df30708dafa7edb23%7C05a0e69a418a47c19c259387261bf991%7C0%7C0%7C638097720042599463%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=BzQh0UsffVZE3eO22b2Xba3p0VMOBZSHfS21FGHXtZM%3D&reserved=0).

NESA holds the only official and up-to-date versions of the NSW Curriculum and syllabus documents. Please visit the NSW Education Standards Authority (NESA) website <https://educationstandards.nsw.edu.au/> and the NSW Curriculum website [https://curriculum.nsw.edu.au/home](https://curriculum.nsw.edu.au/).

[Software Engineering 11–12 Syllabus](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/overview) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2022.

[Software Engineering 11–12 Course Specifications](https://curriculum.nsw.edu.au/learning-areas/tas/software-engineering-11-12-2022/overview#software-engineering-course-specifications-software_engineering_11_12_2022) © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales, 2022.

BootstrapMade (2024) [*Bootstrap Website Templates*](https://bootstrapmade.com/website-templates/), BootstrapMade website, accessed 5 August 2024.

Bulma (n.d.) [*Modern CSS Framework*](https://bulma.io/) [website], accessed 5 August 2024.

Carter A (9 February 2020) [‘The Dramatic History of the Web Browser’ [video]](https://www.youtube.com/watch?v=KziWxegPrww), *d*otnetsheff, YouTube, accessed 5 August 2024.

CertBros (2020) [‘IP Addresses Explained | Cisco CCNA 200-301’ [video]](https://www.youtube.com/watch?v=LIzTo6e4FgY), CertBros, YouTube, accessed 5 August 2024.

Codecademy (2024) [Back-End Web Architecture](https://www.codecademy.com/article/back-end-architecture) [website], accessed 5 August 2024.

Collins T (2023) ‘[Top Web Development Tools in 2023](https://www.browserstack.com/guide/web-development-tools)’, Guide, BrowserStack website, accessed 5 August 2024.

DW Shift (4 August 2021) [‘Tim Berners-Lee: How This Guy Invented the World Wide Web 30 Years Ago’ [video]](https://www.youtube.com/watch?v=GUrDI6OkJfU), DW Shift, YouTube, accessed 5 August 2024.

Embrace The Red (13 November 2020) [‘Web Application Security Fundamentals (must know basics for developers, testers and hackers)’ [video]](https://www.youtube.com/watch?v=-7OX58nHPb8), Embrace The Red, YouTube, accessed 5 August 2024.

Execeratics (2016–2017) [*LF and CRLF online converter*](https://app.execeratics.com/LFandCRLFonline/?l=en), Execeratics website, accessed 5 August 2024.

Federal Register of Legislation (1988) [*Privacy Act (1988)*](https://www.legislation.gov.au/C2004A03712/latest/versions), Australian Government website, accessed 10 April 2024.

Fireship (7 July 2021) [‘DNS Explained in 100 Seconds’ [video]](https://www.youtube.com/watch?v=UVR9lhUGAyU), Fireship, YouTube, accessed 5 August 2024.

FollowAndrew (28 September 2019) [‘What is HTTP? How the Internet Works’ [video]](https://www.youtube.com/watch?v=wW2A5SZ3GkI), FollowAndrew, YouTube, accessed 5 August 2024.

Foundation (1998–2024) [*Foundation: The most advanced responsive front-end framework in the world*](https://get.foundation/) [website], accessed 5 August 2024.

freeCodeCamp.org (9 March 2023) [‘Harvard CS50’s Web Programming with Python and JavaScript – Full University Course’ [video]](https://www.youtube.com/watch?v=vzGllw18DkA), freeCodeCamp.org, YouTube, accessed 5 August 2024.

freeCodeCamp.org (12 April 2023) [‘Bash Scripting Tutorial for Beginners’ [video]](https://www.youtube.com/watch?v=tK9Oc6AEnR4), freeCodeCamp.org, YouTube, accessed 5 August 2024.

FSF (Free Software Foundation) (2022) ‘[Gnu General Public License](https://www.gnu.org/licenses/licenses.html#:~:text=general%20licensing%20help-,The%20GNU%20General%20Public%20License,-The%20GNU%20General)’, GNU Operating System, accessed 19 August 2024.

Google LLC (2016) ‘[Light House: Overview](https://developer.chrome.com/docs/lighthouse/overview/)’, Docs, Chrome for Developers website, accessed 5 August 2024.

ISO (International Organization for Standardization) (2022) [*ISO/IEC 27001:2022: Information security, cybersecurity and privacy protection – Information security management systems – Requirements*](https://www.iso.org/standard/27001), ISO website, accessed 10 April 2024.

Jones B (2024) ‘[Programming\_for\_the\_Web\_Task\_Source](https://github.com/TempeHS/Programming_For_The_Web_Task_Source)’, TempeHS (Tempe High School), GitHub website, accessed 5 August 2024.

Joshi M (2023) ‘[A detailed guide on JavaScript Web Development](https://www.browserstack.com/guide/javascript-web-development)’, Guide, BrowserStack website, accessed 5 August 2024.

—— (2023) ‘[Angular vs React vs Vue: Core Differences](https://www.browserstack.com/guide/angular-vs-react-vs-vue)’, Guide, BrowserStack website, accessed 5 August 2024.

—— (2024) ‘[How to Create a Website using HTML and CSS](https://www.browserstack.com/guide/build-a-website-using-html-css)’, Guide, BrowserStack website, accessed 5 August 2024.

Katz Y (2011–2019) [*Handlebars*](https://handlebarsjs.com/) [website], accessed 5 August 2024.

LaunchCode (4 April 2017) [‘Introduction to Object-Relational Mapping’ [video]](https://www.youtube.com/watch?v=dHQ-I7kr_SY), LaunchCode, YouTube, accessed 7 August 2024.

LearnWebCode (11 February 2010) [‘Learn HTML – Your First Web Page (For Absolute Beginners)’ [video]](https://www.youtube.com/watch?v=BBcq9abR9Es), Learn WebCode*,* YouTube, accessed 5 August 2024.

LearnWebCode (12 February 2010) [‘Learn CSS (Tutorial) – Basics + Selectors (For Absolute Beginners)’ [video]](https://www.youtube.com/watch?v=Awmx8XlgcaE), LearnWebCode, YouTube, accessed 5 August 2024.

Lesics (30 May 2019) [‘How does the INTERNET work? | ICT #2’ [video]](https://www.youtube.com/watch?v=x3c1ih2NJEg), Lesics, YouTube, accessed 5 August 2024.

Lucid Software Inc (2024) ‘[Decision tree diagram maker](https://www.lucidchart.com/pages/examples/decision-tree-maker?a=0)’, *Diagram center*, Lucidchart website, accessed 19 August 2024.

—— (2024)[*How to Make a Decision Tree Diagram*](https://www.lucidchart.com/pages/how-to-make-a-decision-tree-diagram), Lucidchart website, accessed 3 April 2024.

—— (2024) [*Lucidchart*](https://www.lucidchart.com/pages/?) [website], accessed 3 April 2024.

Microsoft Visual Studio Marketplace (2024) [*Extensions for Visual Studio Code*](https://marketplace.visualstudio.com/vscode), Microsoft Visual Studio Marketplace website, accessed 7 August 2024.

Miro (2024)[*Data Flow Diagram (DFD) Tool*](https://miro.com/diagramming/data-flow-diagram/), Miro website, accessed 3 April 2024.

Miro (2024)[*What is a Data Flow Diagram? Examples, Symbols, and Use Cases*](https://miro.com/diagramming/what-is-a-data-flow-diagram/), Miro website, accessed 3 April 2024.

Mozilla Corporation (1998–2024) ‘[Progressive web apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps?ref=arctype.com)’, References, MDN Web Docs website, accessed 7 August 2024.

Mustache (n.d.) [*Mustache Manual GitHub*](https://mustache.github.io/) [website], accessed 5 August 2024.

Nduta A (3 April 2023) ‘[A Brief History of Open Source](https://www.freecodecamp.org/news/brief-history-of-open-source/)’ freeCodeCamp news, accessed 5 August 2024.

Nisbet G (9 August 2014) [‘IP addresses. Explained.’ [video]](https://www.youtube.com/watch?v=7_-qWlvQQtY), George Nisbet, YouTube, accessed 5 August 2024.

NSW Department of Education (2024) ‘[Connecting learning](https://education.nsw.gov.au/teaching-and-learning/curriculum/explicit-teaching/explicit-teaching-strategies/connecting-learning)’, Explicit teaching strategies, NSW Department of Education website, accessed 5 August 2024.

NSW Department of Education (2024) [*Digital Learning Selector*](https://app.education.nsw.gov.au/digital-learning-selector/), NSW Department of Education, website, accessed 5 August 2024.

NSW Legislation (1998) [*NSW Privacy and Personal Information Act (1998) No 133*](https://legislation.nsw.gov.au/view/whole/html/inforce/current/act-1998-133#statusinformation), NSW Legislation website, accessed 10 April 2024.

O'Connor R, de Leastar E and Mullally B (hosts) (09 November 2018) ‘[Beginner's Guide To Web Development](https://podfollow.com/the-machine/episode/0be5037a4b40d3d1c86d4e5a8e89178503162a51/view)’ [podcast], The Machine: A computer science education podcast, podfollow, accessed 5 August 2024.

O'Connor R, Frisby R and Freeman-Gater A (hosts) (12 October 2019) [‘How The Internet Works](https://podfollow.com/the-machine/episode/a513be7ecc8ddcabbdc08a6578b0dce6c55bfc47/view)’ [podcast], The Machine: A computer science education podcast, podfollow, accessed 5 August 2024.

Ostrovsky D and NDC Conferences (3 July 2023) [‘Understanding Big Data for Software Engineers – David Ostrovsky – NDC Oslo 2023’ [video]](https://www.youtube.com/watch?v=NXjSJYyByzE), NDC Conferences, YouTube, accessed 5 August 2024.

PowerCert Animated Videos (28 November 2020) [‘Traceroute (tracert) Explained – Network Troubleshooting’ [video]](https://www.youtube.com/watch?v=up3bcBLZS74), PowerCert Animated Videos, YouTube, accessed 5 August 2024.

Pug (n.d.) [*Getting Started: Installation*](https://pugjs.org/api/getting-started.html#installation), GitHub website, accessed 5 August 2024.

Ray K (20 December 2021) ‘[Everything You Need To Know About Web Application Architecture](https://medium.com/weekly-webtips/everything-you-need-to-know-about-web-application-architecture-57d94ff44c1b)’, Medium, accessed 5 August 2024.

Refsnes Data (1999–2024) [*MongoDB Tutorial*](https://www.w3schools.com/mongodb/index.php), W3Schools website, accessed 5 August 2024.

—— (1999–2024) [*MySQL Tutorial*](https://www.w3schools.com/mysql/default.asp), W3Schools website, accessed 5 August 2024.

—— (1999–2024) [*Node.js Tutorial*](https://www.w3schools.com/nodejs/default.asp), W3Schools website, accessed 5 August 2024.

Steele C (27 February 2021) ‘[30 Years of Browsers: A Quick History](https://au.pcmag.com/browsers/85834/30-years-of-browsers-a-quick-history)’, PCMag, accessed 5 August 2024.

Sourojit D (2024) ‘[Web Application Development: Process, Tools, & Examples](https://www.browserstack.com/guide/web-application-development-guide)’, Guides, BrowserStack website, accessed 5 August 2024.

SuperSimpleDev (20 October 2021) [‘Backend web development – a complete overview’ [video]](https://www.youtube.com/watch?v=XBu54nfzxAQ), SuperSimpleDev, YouTube, accessed 5 August 2024.

TutorialBrain (20 April 2022) [‘How to Run SQLITE in Visual Studio Code’ [video]](https://www.youtube.com/watch?v=JrAiefGNUq8), TutorialBrain, YouTube, accessed 5 August 2024.

W3C (World Wide Web Consortium) (2014) ‘[Tutorials](https://www.w3.org/WAI/tutorials/)’, WAI (Web Accessibility Initiative): Design and Develop Overview, W3C website, accessed 5 August 2024.

W3C (2024) ‘[Web Accessibility Evaluation Tools List](https://www.w3.org/WAI/test-evaluate/tools/list/)’, WAI: Evaluation Tools Overview, W3C website, accessed 5 August 2024.

Web Dev Simplified (25 August 2019) [‘What Is React And Why You Need To Know It’ [video]](https://www.youtube.com/watch?v=1wZoGFF_oi4), Web Dev Simplified, YouTube, accessed 5 August 2024.

Wilson C (27 May 2024) ‘[Bash cheat sheet: Top 25 commands and creating custom commands](https://www.educative.io/blog/bash-shell-command-cheat-sheet)’, Educative blog, accessed 7 August 2024.

Wordpress (n.d.) [*Wordpress*](https://wordpress.com/) [website], accessed 5 August 2024.

**© State of New South Wales (Department of Education), 2024**

The copyright material published in this resource is subject to the *Copyright Act 1968* (Cth) and is owned by the NSW Department of Education or, where indicated, by a party other than the NSW Department of Education (third-party material).

Copyright material available in this resource and owned by the NSW Department of Education is licensed under a [Creative Commons Attribution 4.0 International (CC BY 4.0) license](https://creativecommons.org/licenses/by/4.0/).

[](https://creativecommons.org/licenses/by/4.0/)

This license allows you to share and adapt the material for any purpose, even commercially.

Attribution should be given to © State of New South Wales (Department of Education), 2024.

Material in this resource not available under a Creative Commons license:

* the NSW Department of Education logo, other logos and trademark-protected material
* material owned by a third party that has been reproduced with permission. You will need to obtain permission from the third party to reuse its material.

**Links to third-party material and websites**

Please note that the provided (reading/viewing material/list/links/texts) are a suggestion only and implies no endorsement, by the New South Wales Department of Education, of any author, publisher, or book title. School principals and teachers are best placed to assess the suitability of resources that would complement the curriculum and reflect the needs and interests of their students.

If you use the links provided in this document to access a third-party's website, you acknowledge that the terms of use, including licence terms set out on the third-party's website apply to the use which may be made of the materials on that third-party website or where permitted by the *Copyright Act 1968* (Cth). The department accepts no responsibility for content on third-party websites.