Software Engineering Stage 6 (Year 12) – sample assessment task 2 notification

Secure software architecture

Contents

[About this resource 3](#_Toc183426657)

[Purpose of resource 3](#_Toc183426658)

[Target audience 3](#_Toc183426659)

[When and how to use 3](#_Toc183426660)

[Task description 4](#_Toc183426661)

[Project description 5](#_Toc183426662)

[Submission details 6](#_Toc183426663)

[Component A 6](#_Toc183426664)

[Component B 6](#_Toc183426665)

[Steps to success 7](#_Toc183426666)

[What is the teacher looking for? 10](#_Toc183426667)

[Marking guidelines 11](#_Toc183426668)

[Student-facing rubric 13](#_Toc183426669)

[Appendix 1 16](#_Toc183426670)

[Integrated project across 2 to 3 focus areas 16](#_Toc183426671)

[Task description 16](#_Toc183426672)

[Submission details 17](#_Toc183426673)

[Steps to success 18](#_Toc183426674)

[What is the teacher looking for? 21](#_Toc183426675)

[Marking guidelines 22](#_Toc183426676)

[Student-facing rubric 25](#_Toc183426677)

[Security summary report scaffold 31](#_Toc183426678)

[User-acceptability testing 34](#_Toc183426679)

[Support and alignment 37](#_Toc183426680)

[References 39](#_Toc183426681)

[Legislation 44](#_Toc183426682)

# About this resource

## Purpose of resource

This sample assessment task notification unpacks how teachers can assess students in the Secure software architecture focus area of Software Engineering Year 12.

## Target audience

This resource can be used to support teachers with effective syllabus implementation.

## When and how to use

The resource can be adapted to suit the context of the school. This is sample assessment task 2 of 4 assessment tasks. Teachers can also refer to the sample scope and sequence and assessment schedule. The task is weighted at 20% and requires students to develop a solution, project documentation and presentation.

# Task description

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

**Outcomes being assessed**:

A student:

* justifies methods used to plan, develop and engineer software solutions **SE-12-01**
* applies structural elements to develop programming code **SE-12-02**
* evaluates practices to safety and securely collect, use and store data **SE-12-04**
* justifies the selection and use of tools and resources to design, develop, manage and evaluate software **SE-12-06**
* designs, develop and implements safe and secure programming solutions **SE-12-07**
* tests and evaluates language structures to refine code **SE-12-08**
* applies methods to manage and document the development of a software project **SE-12-09**

[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.

**Suggested weighting**: 20%

Your client, ‘The Unsecure PWA Company’, has engaged you as a software engineering security specialist to provide expert advice on the security and privacy of their application. This progressive web app is currently in the testing and debugging phase of the software development lifecycle.

Access to the [Unsecure progressive web app (PWA)](https://github.com/TempeHS/The_Unsecure_PWA).

# Project description

Your client, ‘The Unsecure PWA Company’,has engaged you as a software engineering security specialist to provide expert advice on the security and privacy of their application. This progressive web app is currently in the testing and debugging phase of the software development lifecycle and can be accessed here: [Unsecure progressive web app (PWA)](https://github.com/TempeHS/The_Unsecure_PWA).

You are to:

* run a range of security tests and scans, along with a white/grey/black box analysis of the application to identify as many security and privacy vulnerabilities as possible
* provide examples of secure code solutions, for example, the application programming interface (API) may include a web-based login component that provides a solution using a [Python and Flask framework](https://github.com/TempeHS/Flask_PWA_Programming_For_The_Web_Task_Source) to allow for two-factor authentication (2FA) and cross-site request forgery (CSRF) protection and session management
* prepare a professionally written security report for your client that includes
* an overview of your approach to the technical analysis
* documentation of the scope of privacy and security issues, including
* security or privacy issues that cannot be mitigated by technical engineering solutions
* security issues that must be tested in the production environment
* security or privacy vulnerabilities you discovered and an impact assessment of each
* recommendations relating to a security and privacy by design approach going forward for ‘The Unsecure PWA Company’
* designing and developing implementations using HTML/CSS/JS/SQL/JSON/Python code and/or web content changes as required to patch each vulnerability you discover.

# Submission details

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

## Component A

A complete security report (scaffolded example in the Appendix of the teacher support resource).

See the [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) for standardised versions of project documentation. These need to be adhered to throughout the course and may form part of examinable content.

## Component B

A zip file of any implemented solution or sample secure code.

Component C

A presentation to the class who are playing the role of the client, ‘The Unsecure PWA Company’, with questions and answers.

## Steps to success

Table 1 – assessment preparation schedule

|  |  |
| --- | --- |
| Steps | What I need to do |
| Test and evaluate the progressive web app (PWA) | 1. Experiment with the unsecure PWA. 2. Investigate how the unsecure PWA is built to see how data is handled. 3. Locate any security weaknesses in the code and how data is handled. 4. Conduct a security audit and vulnerability assessments of the unsecure PWA using special tools and tests to find any problems. 5. Document the processes you followed and the vulnerabilities or issues you discovered. This could include:  * browser developer tool reports (Google Lighthouse and Edge Application) * black box test the app for vulnerabilities and privacy issues * grey box test the app * code review/white box asses the application * use of third-party tools like [Pentest-Tools.com](https://pentest-tools.com) or [Zed Attack Proxy (ZAP)](https://www.zaproxy.org/). |
| Designing software | 1. Write a requirements definition for the client. 2. Identify the user specifications for the solution. 3. Apply the fundamental software development steps to develop secure code. 4. Identify any problems that cannot be fixed by changing the code, such as how users behave or how data is managed.   Consider:   * What will be different when the application is in a production environment? * What might users do that you or the ‘The Unsecure PWA Company’ cannot control that could be a vulnerability or a privacy issue? * What can’t be tested? |
| Developing secure code | 1. Design, develop and implement code that changes the PWA's code and settings to fix the security issues found, like checking user inputs, encrypting data and controlling accessibility. 2. Use appropriate HTML/CSS/JS/SQL/JSON/Python code and web content changes to provide a close-to-industry standard solution that fully or near fully mitigates security and privacy vulnerabilities. 3. Apply strategies to manage the security of programming code. 4. Test and evaluate the security and resilience of the software. |
| Impact of safe and secure software development | 1. Describe the benefits of developing secure software to your client ‘The Unsecure PWA Company’. 2. Provide expert advice to the ‘The Unsecure PWA Company’ on the privacy and security of their progressive web app (PWA). 3. List all the security or privacy issues found and explain their impact if someone took advantage of them by providing an impact assessment of each. 4. Present your solution and security report to the client. 5. Explain the benefits of implementing safe and secure development practices to an enterprise. 6. Explain the social, ethical and legal issues that affect people and enterprises, resulting from the development and implementation of safe and secure software. |

# What is the teacher looking for?

Students have:

* performed a thorough analysis of the documentation, code, user interfaces and user experiences to discover the most vulnerabilities and privacy issues, and identified and explained the most out of scope vulnerabilities
* documented evidence of the effective application of a variety of automated tools and brute force attacks to discover the most vulnerabilities
* investigated how the PWA is built to find any security weaknesses in the code and how data is handled
* found any problems that cannot be fixed by changing the code, like how users behave or how data is managed
* suggested ways to build future PWAs with appropriate security and privacy from the start
* made changes to the PWA's code and settings to fix the security issues found, like checking user inputs, encrypting data and controlling who can access what
* used appropriate HTML/CSS/JS/SQL/JSON/Python code and web content changes to provide a close-to-industry standard solution that fully or near fully mitigates security and privacy vulnerabilities
* completed the security report and presented findings and a more secure PWA to the client during a class presentation.

# Marking guidelines

Table 2 – assessment marking guidelines

|  |  |
| --- | --- |
| ****Grade**** | ****Marking guideline descriptors**** |
| ****A**** | The student:   * demonstrates an extensive understanding of the steps used by programmers when designing software * documents the software development steps thoroughlyin a highly developed security report * develops highly effective algorithms that demonstrate the logic required for a software solution * uses highly effective analysis in the development of computing solutions using the <python> programming language and <Flask> framework.\* |
| ****B**** | The student:   * demonstrates a thorough understanding of the steps used by programmers when designing software * documents these in a well-developed security report * develops effective algorithms that demonstrate the logic required for a software solution * uses effective analysis in the development of computing solutions using the <python> programming language and <Flask> framework.\* |
| ****C**** | The student:   * demonstrates a sound understanding of the steps used by programmers when designing software * documents some of these in a sound security report * develops sound algorithms to demonstrate the logic required for a software solution * uses appropriate computing solutions using the <python> programming language and <Flask> framework.\* |
| ****D**** | The student:   * demonstrates a basic understanding of the steps used by programmers when designing software * documents some of these in the security report * develops basic algorithms to demonstrate some of the logic required for a software solution * implements basic computing solutions using the<python> programming language and <Flask> framework.\* |
| ****E**** | The student:   * identifies the requirements for the documentation and production of a software solution. |

\* Insert relevant <programming language> and <framework>.

# Student-facing rubric

Table 3 – rubric for Secure software architecture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ****Submission criteria**** | ****Limited**** | ****Basic**** | ****Sound**** | ****High**** | ****Outstanding**** |
| ****Criteria 1****  ****Designing software**** | Labels the fundamental software development steps to develop secure code. | Identifies the fundamental software development steps to develop secure code. | Applies and outlines the fundamental software development steps to develop secure code. | Applies and explains fundamental software development steps to develop secure code. | Applies and justifies the use of all fundamental software development steps to develop secure code. |
| ****Criteria 2****  ****Developing solutions with code**** | Completes missing code to solve the task.  Documents use of data structures and errors. | Modifies code to solve the task.  Code includes some use of the control structures, 2FA, CSRF, some use of data structures and parameter passing.  Identifies their use of at least one software debugging tool.  Correctly describes at least one error experienced in the coding of their solution. | Creates code to solve the task.  Code includes some use of the control structures, 2FA, CSRF, some use of data structures and parameter passing.  Defines and documents their use of at least one software debugging tool.  Describes at least one error experienced in the coding of their solution. | Creates and documents secure code to solve the task.  Code includes correct use of most parameter passing, most of the control structures, 2FA, CSRF and use of data structures and files.  Correctly describes a range of errors experienced in the coding of their solution. | Creates, documents, and justifies the use of efficient and secure code to solve the task.  Code includes correct use of parameter passing, all the control structures, 2FA, CSRF, effective use of data structures and files.  Correctly describes, in detail, a range of errors experienced in the coding of their solution and how they were corrected. |
| ****Criteria 3****  ****Impact of safe and secure software development**** | Labels the session management technique used in their project.  Identifies a ‘Data dictionary’.  Identifies the data structures used in their project.  Identifies a security strategy used in programming code. | Defines the session management technique used in their project.  Fills in an incomplete ‘Data dictionary’.  Defines the data structures used in their project.  Defines a security strategy used in programming code. | Creates a ‘Data dictionary’ for their project.  Defines the data structures used in their project.  Discusses the session management technique used in their project.  Defines a security strategy for protecting their programming code. | Creates a comprehensive ‘Data dictionary’ for use with their project.  Explains the data structures and discusses their use in their project.  Explains their selection of session management technique used in their project.  Explains appropriate security strategies for their programming code. | Presents a secure PWA and a comprehensive security report to the client explaining the benefits to their enterprise of the implementation of safe and secure development practices.  Justifies the use of a comprehensive ‘Data dictionary’ for their project.  Justifies the data structures used in their project.  Justifies how their file is protected against vulnerabilities.  Justifies their selection of session management technique used in their project.  Justifies appropriate security strategies for their programming code. |

# Appendix 1

## Integrated project across 2 to 3 focus areas

### Task description

Create and document a progressive web app (PWA) that catalogues products with an authentication-protected web front-end to add new catalogue items.

You are to design and document a software solution for a secure progressive web app developed in a high-level general-purpose programming language.

The purpose of the application is to catalogue something students themselves are interested in and have a web front-end that requires authentication so they can add new items.

**Specifications**

* Add items page protected by two-factor authentication (2FA)
* Allow users to sort and filter the public catalogue
* Appropriate session management
* An intuitive web-based interface that displays directions and responses to the user’s requests
* The software needs to include the use of appropriate data structures
* Defensive data input handling and sanitisation practices through an API

# Submission details

[Provide any important details regarding submission. It may be about word limits, time limits, procedures if absent when due, format of task submission, where to submit and so on.]

Students should be fully aware of the school assessment procedures for their year group.

Suggested weighting: NN%

# Steps to success

Table 4 – assessment preparation schedule

|  |  |
| --- | --- |
| ****Steps**** | ****What I need to do**** |
| ****Software development****  ****Apply the fundamental software development steps in relation to their project**** | * Write a requirements definition for the given real-world problem * Identify the user specifications for the chosen catalogue |
| ****Set up the development environment**** | * Install the required extensions and packages * Create the required folder structure and files |
| ****Database design**** | * Develop a schema for the database, including field names, primary and foreign keys and joins |
| ****Designing algorithms**** | * Develop a structured algorithm using pseudocode or flowcharts, including the use of subprograms, passing parameters and accessing files. Algorithms should include sequence, selection, iteration and sub programs * Develop SQL queries to access the database to filter, insert and retrieve relevant data |
| ****Designing a user interface (UI)**** | * Develop a storyboard for the website that considers accessibility and inclusivity, and applies appropriate design, user interface (UI) and user experience (UX) principles |
| ****Data for software engineering**** | * Identify and discuss how the implemented source code minimises vulnerabilities such as: * broken authentication and session management * cross-site scripting (CSS) and cross-site request forgery (CSRF) * invalid forwarding and redirecting * SQL injections. * Create a ‘Data dictionary’ for use with your project |
| ****Developing solutions with code**** | * Develop and construct your website using HTML and CSS * Convert your algorithm into code using: * control structures * data and structures * standard modules * functions (including parameter passing) * database commands (inserting, retrieving and querying). * Document and implement a 2FA method * Document and implement session management * Document and implement a data handling API * Document and implement appropriate data structures and file handling that supports data storage * Document and implement an appropriate hashing algorithm for securely storing passwords in the database * Complete user-accessibility testing using a range of appropriate scenarios |
| ****Write a security summary report**** | * Summarise what processes, technical solutions or design principles you applied at each step of the software development lifecycle * Summarise how you have applied privacy by design in your solution * Summarise how you have minimised vulnerabilities in user action controls, including * broken authentication and session management * cross-site scripting (XSS) and cross-site request forgery (CSRF) * invalid forwarding and redirecting * race conditions. |

# What is the teacher looking for?

This task will require students to choose an appropriate catalogue item (for example, music, concerts, movies, actors, recipes, games, restaurants, sneakers, food, art and clothes).

They will then design, develop and document a software solution coded in HTML, CSS, JavaScript and SQL programming language and utilise the Node.js or Python Flask frameworks to allow for secure communication between the website and database.

# Marking guidelines

Table 2 – assessment marking guidelines

|  |  |
| --- | --- |
| ****Grade**** | ****Marking guideline descriptors**** |
| ****A**** | The student:   * demonstrates an extensive understanding of the steps used by programmers when designing private and secure software by design * develops highly effective algorithms to demonstrate the logic required for a software solution * develops highly effective software solutions using HTML, CSS and the JavaScript programming language, and Node.js and/or Python Flask frameworks * demonstrates an extensive understanding of designing and creating a user interface and user experience to meet accessibility requirements * develops an appropriate database schema, and applies highly effective and secure SQL queries to access the database * demonstrates an extensive understanding and ability to test and debug a software solution. |
| ****B**** | The student:   * demonstrates a thorough understanding of the steps used by programmers when designing secure software while addressing privacy issues * develops effective algorithms to demonstrate the logic required for a software solution * develops effective software solutions using HTML, CSS and the JavaScript programming language, and Node.js and/or Python Flask frameworks * demonstrates a thorough understanding of designing and creating a user interface and user experience to meet accessibility requirements * develops an appropriate database schema, and effective and secure SQL queries to access the database * demonstrates a thorough understanding and ability to test and debug a software solution. |
| ****C**** | The student:   * demonstrates a sound understanding of the steps used by programmers when designing software with evidence of addressing privacy and security issues * develops sound algorithms that demonstrate the logic required for a software solution * develops sound software solutions using HTML, CSS and the JavaScript programming language, and Node.js and/or Python Flask frameworks * demonstrates a sound understanding of designing and creating a user interface and user experience to meet accessibility requirements * develops a schema that demonstrates a sound understanding of database construction and SQL queries * demonstrates a sound ability to test and debug a software solution. |
| ****D**** | The student:   * demonstrates a basic understanding of the steps used by programmers when designing software * develops basic algorithms to demonstrate the logic required for a software solution * develops a basic software solution using HTML, CSS and the JavaScript programming language, and Node.js and/or Python Flask frameworks * demonstrates a basic understanding of designing and creating a user interface and user experience to meet accessibility requirements * develops a schema and SQL queries that demonstrates a basic understanding of databases * demonstrates a basic ability to test and debug a software solution. |
| ****E**** | The student:   * identifies the requirements for the documentation and production of a software solution. |

# Student-facing rubric

Table 3 – student-facing assessment rubric

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ****Criteria**** | ****Limited**** | ****Basic**** | ****Sound**** | ****High**** | ****Outstanding**** |
| ****Criteria 1****  ****Apply the fundamental software development steps in relation to their project**** | Labels the fundamental software development steps in their project. | Identifies the fundamental software development steps in their project. | Refers to the fundamental software development steps when outlining secure architecture or privacy by design. | Explains and applies fundamental software development steps when implementing secure architecture or privacy by design. | Comprehensively explains and applies the fundamental software development steps to implement secure architecture and privacy by design in their project. |
| ****Criteria 2****  ****Directory structure**** | Creates a single directory. | Creates some files and folders to form a basic structure. | Creates and labels files and folders to form a suitable structure. | Creates and labels files and folders into an appropriate and secure structure using appropriate naming conventions. | Creates and labels files and folders into a highly appropriate and secure structure using highly appropriate naming conventions. |
| ****Criteria 3****  ****Database design**** | Identifies field names or demonstrates a limited understanding of a schema.  Identifies a ‘Data dictionary’. | Develops a schema which contains some field names.  Provides an incomplete ‘Data dictionary’. | Develops a correct schema for the database, including field names, and primary or foreign keys.  Creates a ‘Data dictionary’ for their project. | Develops a relevant schema for the database, including field names, primary and foreign keys and joins.  Creates a correct ‘Data dictionary’ for use with their project. | Designs and develops a highly relevant schema for the database, including appropriate field names, primary and foreign keys and joins.  Creates a comprehensive ‘Data dictionary’ for use with their project. |
| ****Criteria 4****  ****Designing algorithms**** | Identifies the control structures in an algorithm.  Completes missing SQL queries to solve a task description. | Develops an algorithm which makes some use of the control structures.  Modifies SQL queries to store or retrieve data. | Develops a correct algorithm that correctly uses all control structures, including sequence, selection, repetition, files, dictionaries, classes and functions.  Develops correct SQL queries to store or retrieve data. | Develops a correct algorithm, detailing the steps required for a solution to their problem. The algorithm makes the correct use of all the control structures, including sequence, selection, repetition, files, dictionaries, classes and functions.  Develops correct SQL queries to store and retrieve data. | Designs, develops and desk checks structured algorithms, documenting the steps required for a solution to their problem. The algorithm makes the correct use of all the control structures, including sequence, selection, repetition, files, dictionaries, classes and functions.  Designs and develops correct SQL queries to store and retrieve data, joining tables on unique keys. |
| ****Criteria 5****  ****Designing a user interface**** | Develops a storyboard for the website that demonstrates a limited understanding of accessibility, inclusivity and design principles. | Develops a storyboard for the website that demonstrates some understanding of accessibility and inclusivity. The student applies some appropriate design principles. | Develops a storyboard for the website that demonstrates an understanding of accessibility and inclusivity. The student applies appropriate design principles. | Develops a storyboard for the website that demonstrates a high level of understanding of accessibility and inclusivity. The student applies a range of appropriate design, user interface (UI) and user experience (UX) principles. | Designs and develops a storyboard for the website that is highly accessible and inclusive, and applies a wide range of appropriate design, user interface (UI) and user experience (UX) principles. |
| ****Criteria 6****  ****Developing solutions with code**** | Produces incomplete code that addresses some of the task description.  Makes elementary connections between design, and security or privacy. | Modifies code to solve the task description.  Code includes some use of control structures, 2FA, a hashing algorithm, some use of databases, data structures and parameter passing.  Implements basic practices to develop secure applications.  Applies basic accessibility in the progressive web app. | Creates code to solve the task description.  Code includes some use of control structures, 2FA, a hashing algorithm, sanitisation of login information, and some use of a database, data structures and parameter passing.  Demonstrates developing skills in implementing secure applications.  Performs accessibility testing. | Creates and documents secure code to solve the task description.  Code includes use of control structures with correct use of SQL, HTML, CSS, JS and Python.  Mostly implement a data handling API, 2FA, hashing algorithm, and use of a database, data structures and files.  Demonstrates a range of appropriate security strategies that secure their application.  Correctly implements accessibility testing using appropriate scenarios. | Creates, documents, and explains efficient and secure code to solve the task description.  Code includes use of control structures and parameter passing with correct use of SQL, HTML, CSS, JS and Python.  Implements a data handling API, 2FA, hashing algorithm, and use of a database, data structures and files.  Implements a range of current secure measures that result in a secure architecture for their application.  Correctly implements accessibility testing using a range of highly appropriate scenarios. |
| ****Criteria 7****  ****Write a**** ****security summary report**** | Identifies some techniques used in their project to protect against vulnerabilities. | Defines the techniques used in their project to protect against vulnerabilities. | Outlines the techniques used in their project to protect against vulnerabilities, including broken authentication and session management, cross-site scripting and cross-site forgery, and invalid forwarding and redirecting. | Explains their selection and implementation of source code to protect against vulnerabilities, including broken authentication and session management, cross-site scripting and cross-site forgery, and invalid forwarding and redirecting. | Justifies their selection and implementation of source code to protect against vulnerabilities, including broken authentication and session management, cross-site scripting and cross-site forgery, and invalid forwarding and redirecting. |

# Security summary report scaffold

Summarise the processes you have followed at each stage of the software development life cycle that contribute to the overall security of your application or privacy of users.

Table 4 – stage of the software development life cycle

|  |  |  |
| --- | --- | --- |
| ****Stage**** | ****Secure architecture planning and design principles applied**** | ****Privacy by design principles applied**** |
| ****Requirements definition**** |  |  |
| ****Determining specifications**** |  |  |
| ****Design**** |  |  |
| ****Development**** |  |  |
| ****Integration**** |  |  |
| ****Testing and debugging**** |  |  |
| ****Installation**** |  |  |
| ****Maintenance**** |  |  |

Table 5 –mitigating against vulnerabilities

|  |  |  |
| --- | --- | --- |
| ****Vulnerability**** | ****Describe the implemented security solutions**** | ****Describe how the solutions have improved application or user security and privacy.**** |
| ****Broken authentication and session management**** |  |  |
| ****Cross-site scripting (XSS) and cross-site request forgery (CSRF)**** |  |  |
| ****Invalid forwarding and redirecting**** |  |  |
| ****Race conditions**** |  |  |

## User-acceptability testing

Performed at the end of the testing phase of the software development lifecycle, the software is tested in the ‘real world’ by the intended audience or business representative. Students need to describe the steps to test and the expected outcome. Then, have a group of students perform the test and validate that the application is acceptable to the users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****Test**** | ****Describe the feature being tested**** | ****Describe the user input or test data**** | ****Describe the pass criteria**** | Test validation |
|  |  |  |  | Tester name:  PASS  FAIL  Observations: |
|  |  |  |  | Tester name:  PASS  FAIL  Observations: |
|  |  |  |  | Tester name:  PASS  FAIL  Observations: |
|  |  |  |  | Tester name:  PASS  FAIL  Observations: |
|  |  |  |  | Tester name:  PASS  FAIL  Observations: |

**Tip**: Write step-by-step, detailed but concise instructions on how to test the feature.

## Support and alignment

**Resource evaluation and support**: all curriculum resources are prepared through a rigorous process. Resources are periodically reviewed as part of our ongoing evaluation plan to ensure currency, relevance and effectiveness. For additional support or advice contact the TAS curriculum team by emailing [TAS@det.nsw.edu.au](mailto:TAS@det.nsw.edu.au).

**Differentiation**: further advice to support Aboriginal and Torres Strait Islander students, English as an additional language or dialect (EAL/D) students, students with a disability and/or additional needs and High Potential and Gifted (HPG) students can be found on the [Planning programming and assessing 7–12](https://education.nsw.gov.au/teaching-and-learning/curriculum/planning-programming-and-assessing-k-12/planning-programming-and-assessing-7-12) webpage. This includes the [Inclusion and differentiation 7–10 advice](https://education.nsw.gov.au/teaching-and-learning/curriculum/planning-programming-and-assessing-k-12/planning-programming-and-assessing-7-12/inclusion-and-differentiation-advice-7-10) webpage.

**Assessment**: further advice to support formative assessment is available on the [Planning programming and assessing 7–12](https://education.nsw.gov.au/teaching-and-learning/curriculum/planning-programming-and-assessing-k-12/planning-programming-and-assessing-7-12) webpage. This includes the [Classroom assessment advice 7–10](https://education.nsw.gov.au/teaching-and-learning/curriculum/planning-programming-and-assessing-k-12/planning-programming-and-assessing-7-12/classroom-assessment-advice-7-10-). For summative assessment tasks, the [Assessment task advice 7–10](https://education.nsw.gov.au/teaching-and-learning/curriculum/planning-programming-and-assessing-k-12/planning-programming-and-assessing-7-12/assessment-task-advice-7-10) webpage is available.

**Consulted with**: Curriculum and Reform and subject matter experts.

**Alignment to software priorities and/or needs**: [School Excellence Policy](https://education.nsw.gov.au/policy-library/policies/pd-2016-0468)

**Alignment to the School Excellence Framework**: this resource supports the [School Excellence Framework](https://education.nsw.gov.au/policy-library/policies/pd-2016-0468) elements of curriculum (curriculum provision) and effective classroom practice (lesson planning, explicit teaching).

**Alignment to Australian Professional Standards for Teachers**: this resource supports teachers to address [Proficient Teacher Standard Descriptors](https://www.nsw.gov.au/education-and-training/nesa/teacher-accreditation/proficient-teacher/standard-descriptors) **3.1.2, 3.3.2, 3.4.2, 5.1.2.**

**NSW Syllabus**: Software Engineering 11–12

**Syllabus outcomes**: SE-12-01, SE-12-02, SE-12-04, SE-12-06, SE-12-07, SE-12-08, SE-12-09

**Author**: TAS, Curriculum Secondary Learners, Curriculum Reform

**Publisher**: State of NSW, Department of Education

**Resource**: sample Assessment task notification

**Related resources**: further resources to support Software Engineering 11–12 can be found on the [TAS curriculum page](https://education.nsw.gov.au/teaching-and-learning/curriculum/tas).

**Professional learning**: relevant professional learning is available through [HSC Professional Learning](https://education.nsw.gov.au/teaching-and-learning/professional-learning/hsc-pl) or on the [TAS curriculum page](https://education.nsw.gov.au/teaching-and-learning/curriculum/tas).

**Creation date: 2024**

**Rights**: © State of New South Wales, Department of Education.

# 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>.

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>.

[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 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.

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

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

Brookhart SM (2018) ‘[Appropriate Criteria: Key to Effective Rubrics](https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2018.00022/full)’, Frontiers in Education, 3(22):1–12, doi:10.3389/feduc.2018.00022, accessed 3 April 2024.

Campos PT (27 April 2020) ‘[AES Implementation in Python](https://medium.com/quick-code/aes-implementation-in-python-a82f582f51c2)’, Medium: Quick Code, accessed 11 April 2024.

CertBros (2 June 2020) [‘IP Addresses Explained | Cisco CCNA 200-301’ [video]](https://www.youtube.com/watch?v=LIzTo6e4FgY), CertBros, YouTube, accessed 5 August 2024.

CESE (Centre for Education Statistics and Evaluation) (2020a) [*What works best: 2020 update*](https://education.nsw.gov.au/about-us/education-data-and-research/cese/publications/research-reports/what-works-best-2020-update), NSW Department of Education, accessed 3 April 2024.

——(2020b) [*What works best in practice*](https://education.nsw.gov.au/about-us/education-data-and-research/cese/publications/practical-guides-for-educators-/what-works-best-in-practice), NSW Department of Education, accessed 3 April 2024.

——(2021) [*Growth goal setting – what works best in practice*](https://education.nsw.gov.au/about-us/education-data-and-research/cese/publications/practical-guides-for-educators/growth-goal-setting), NSW Department of Education, accessed 3 April 2024.

Codecademy Team (2024) ‘[Back-End Web Architecture](https://www.codecademy.com/article/back-end-architecture)’, Codecademy articles, 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.

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

Delgado C (9 July 2021) ‘[How to create MD5 hashes in JavaScript](https://ourcodeworld.com/articles/read/1547/how-to-create-md5-hashes-in-javascript)’, Our Code World articles,accessed 11 April 2024

dotnetsheff (9 February 2020) [‘The Dramatic History of the Web Browser’ [video]](https://www.youtube.com/watch?v=KziWxegPrww), dotnetsheff, YouTube, 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 converter online*](https://app.execeratics.com/LFandCRLFonline/?l=en), Execeratics website, accessed 5 August 2024.

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

Fisher D and Frey N (2009) [‘Feed Up, Back, Forward](https://ascd.org/el/articles/feed-up-back-forward)’, ASCD (Association for Supervision and Curriculum Development): Educational Leadership, 67(3)20–25, accessed 3 April 2024.

FollowAndrew (28 September 2019) [‘What is HTTP? How the Internet Works! #1’ [video]](https://www.youtube.com/watch?v=wW2A5SZ3GkI), FollowAndrew, YouTube, accessed 5 August 2024.

Foundation (1998–2024) [*Foundation Framework*](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.

Free Software Foundation, Inc (2007) ‘[GNU General Public License Version 3](https://www.gnu.org/licenses/gpl-3.0.txt)’, Licenses, GNU Operating System website, accessed 7 August 2024.

GeeksforGeeks (2023) ‘[MD5 hash in Python](https://www.geeksforgeeks.org/md5-hash-python/)’, GeeksforGeeks articles, accessed 11 April 2024.

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

Griffin P (2017) Assessment for Teaching, Cambridge University Press, Port Melbourne, Victoria.

Hattie J and Timperley H (2007) ‘The Power of Feedback’, Review of Educational Research, 77(1): 81–112, doi:10.3102/003465430298487.

ISO (International Organization for Standardization) (2022) [*ISO/IEC 27001:2022 – Information security management systems*](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, GitHub website, accessed 5 August 2024.

Joshi M (11 May 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.

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

——(7 June 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), LearnWebCode*,* 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.

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

——(2024) [*Flowchart Symbols and Notation*](https://www.lucidchart.com/pages/flowchart-symbols-meaning-explained?a=0), Lucidchart website, accessed 3 April 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.

MDN contributors (2023) ‘[Progressive web apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps?ref=arctype.com)’, References: Web technology for developers, MDN website, accessed 7 August 2024.

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

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

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

Mustafic A (26 July 2021) ‘[PGP key pair generation and encryption and decryption examples in Python 3](https://medium.com/@almirx101/pgp-key-pair-generation-and-encryption-and-decryption-examples-in-python-3-a72f56477c22)’, Medium, accessed 11 April 2024.

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

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.

O'Connor R (host) (9 November 2018) ‘[Beginner's Guide To Web Development](https://podfollow.com/the-machine/episode/0be5037a4b40d3d1c86d4e5a8e89178503162a51/view)’ [podcast], The Machine: A computer science education podcast, SETU (South East Technology University), accessed 5 August 2024.

O'Connor R (11 October 2019) ‘[How The Internet Works](https://podfollow.com/the-machine/episode/a513be7ecc8ddcabbdc08a6578b0dce6c55bfc47/view)’ [podcast], The Machine: A computer science education podcast, SETU (South East Technology University), accessed 5 August 2024.

OpenPGP (2024) [*openpgpjs*](https://github.com/openpgpjs/openpgpjs), GitHub website, accessed 11 April 2024.

Panadero E and Jonsson A (2013) ‘[The use of scoring rubrics for formative assessment purposes revisited: A review](https://www.sciencedirect.com/science/article/abs/pii/S1747938X13000109?via%3Dihub)’, Educational Research Review, 9:129–144, doi:10.1016/j.edurev.2013.01.002, accessed 3 April 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), Pug 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: Webtips, accessed 5 August 2024.

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

Sherrington T (2019) Rosenshine’s Principles in Action, John Catt Educational Limited, Melton, Woodbridge.

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

State of New South Wales (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.

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 Australia, 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.

Thomas J (n.d.) [*Bulma: The Modern CSS Framework*](https://bulma.io/) [website], 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) (2021) ‘[Web Accessibility Evaluation Tools List](https://www.w3.org/WAI/test-evaluate/tools/list/)’, WAI (Web Accessibility Initiative), W3C website, accessed 5 August 2024.

W3C (2023) ‘[Tutorials](https://www.w3.org/WAI/tutorials/)’, WAI, W3C website, accessed 5 August 2024.

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

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

W3Schools (1999–2024) [*MongoDB Tutorial*](https://www.w3schools.com/mongodb/index.php), W3Schools 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.

## Legislation

Privacy Act 1988

Privacy and Personal Information Protection Act 1998 No 133

**© 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.