Mullumbimby High School
Assessment Task

Student Name:	

Task Name:	Progressive Web Application		
Year:	12	**Faculty:**	TAS
Subject:	Software Engineering	**Teacher:**	Steedman
Date Assigned:	12/11/2026	**Date Due:**	19/12/2026
Weighting:	20%	**Total Mark:**	___ / 150

Outcomes to be Assessed:

SE-11-01 - describes methods used to plan, develop and engineer software solutions
SE-11-02 - explains how structural elements are used to develop programming code
SE-11-04 - applies safe and secure practices to collect, use and store data
SE-11-06 - applies tools and resources to design, develop, manage and evaluate software
SE-11-08 - applies language structures to refine code
SE-11-09 - manages and documents the development of a software project

Task Description:

In this assessment task, you will demonstrate your understanding of web programming concepts and apply your software engineering skills. The task consists of two parts: a theory section (50 marks) covering various aspects of web technologies, and a practical component (100 marks) where you'll develop a Progressive Web Application (PWA) using Flask, SQLite, and modern web development techniques. This comprehensive task will challenge you to apply your Python programming knowledge in a web context, integrating databases, user interface design, and PWA features.

Submission Instructions:

Hand in:

1. This completed workbook
2. Your project via GitHub Classroom

Absence / Misadventure Instructions:	Academic Integrity:	Acknowledgement:
All Stage 6 students have been issued with an Assessment Policy Handbook, which outlines the procedures relating to absence/illness/misadventure. This handbook can also be found on the school website. Students must submit an illness/misadventure form to the Head Teacher Senior Studies (Mrs Elliott) for foreseeable absences/misadventures before the task due date and immediately on return to school for unforeseeable absences / misadventures. The following penalties for late submission without an acceptable reason will apply: one day late - 20% deducted, two days late - 40% deducted. Three or more days late will be awarded zero.	Any instances of academic dishonesty may be interpreted as a non-attempt (failing grade). Please refer to the assessment policy booklet for more information. To ensure academic integrity, teachers may require students to submit their work using Google Docs or other platforms with version history tracking (or some other means of evidence of authorship). This enables teachers to verify that the content has been created by the student and not generated by an AI tool.	CLICK HERE OR SCAN

Part 1 - Theory (50)

Web Programming Applications (8):

For each of the following web programming applications provide:

- A description and key features **(1)**
- At least two examples of real-world websites **(1)**

Answer everything in your own words after doing some research. Include two references below.

Type	Description / Key Features	Examples
Static Website		
Interactive Website		
Ecommerce Website		
PWA (Progressive Web Application)		
References:		

Data Transfer (11):

1. Describe two key features of data packets and explain their importance in internet data transfer. **(3)**

2. Describe the structure of IPv4 addresses and explain their role in internet communication. **(2)**

3. Describe the primary function of DNS and explain its importance in web browsing. **(2)**

4. *Sarah is working on a research project and needs to access a specific webpage. She opens her web browser and types "www.researchdatabase.edu" into the address bar. She presses Enter, and after a brief moment, the webpage loads on her screen, displaying a search interface for academic papers.*

Explain the entire process of how Sarah's computer retrieves this webpage, from the moment she presses Enter to when the page appears on her screen. Your explanation must incorporate and demonstrate how data packets, IP addresses, and DNS work together to facilitate this data transfer on the internet. **(4)**

Web Protocols and Ports (15)

1. Complete the following table by describing the purpose of each protocol and listing its standard port number(s). **(1 mark each)**

Protocol	Purpose	Port(s)
HTTP		
HTTPS		
TCP/IP		
DNS		
FTP		
SFTP		
SSL/TLS		
SMTP		
POP3		

2. Examine the screenshot of the Dev Tools provided:

```
General
request URL:        https://httpbin.org/post
request Method:     POST
status Code:        200 OK
remote Address:     54.159.225.35:443
Response Headers
content-length:     1572
content-type:       application/json
date:               Tue, 01 Oct 2024 11:39:17 GMT
server:             gunicorn/19.9.0
Request Headers
:authority:         httpbin.org
:method:            POST
:path:              /post
:scheme:            https
content-length:     189
content-type:       application/x-www-form-urlencoded
origin:             https://httpbin.org
referer:            https://httpbin.org/forms/post
user-agent:         Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/129.0.0.0 Safari/537.36
Form Data
custname:           Dave
custtel:            123456789
custemail:          dave@email.com
size:               medium
topping:            cheese
delivery:           13:15
comments:           Deliver to Mullumbimby High School, Room A12
```

Explain the **client-server model** and the **request-response cycle** in the context of **HTTP**, using the provided screenshot as an example. In your answer:

- Describe the roles of the client and server in this transaction
- Explain the request-response cycle, including what happens on both the client and server sides
- Identify and explain the significance of the HTTP method used
- Analyse the status code in the response and explain its meaning
- Discuss the request payload and how it relates to the form submission **(6 marks)**

Securing the Web (8)

1. Differentiate between authentication and authorisation in the context of web security. Provide an example of each. **(2 marks)**

2. Explain the purpose of SSL certificates and describe how they contribute to web security. **(2 marks)**

3. Describe the process of password hashing and explain why it's preferred over storing plain text passwords. In your explanation, include the concepts of cipher text, salting and hash values, and discuss how this method contributes to safer user authentication. **(4 marks)**

Big Data (8)

1. Explain what "big data" is and describe the three V's (Volume, Velocity, and Variety) that characterise it. Provide an example for each V. **(4 marks)**

2. Explain how big data, including data mining and metadata, is used in web applications to enhance user experience. Provide two examples of real-world applications that utilise these techniques to personalise interactions.**(4 marks)**

Marking Rubric - Theory Tasks

Requirements & Specifications					
Criteria	Basic	Limited	Effective	Highly Effective	**Mark**
Web Programming Applications	Provides basic descriptions for 1-2 application types Lists 1-2 examples Minimal or no references provided	Provides basic descriptions for 3 application types Lists some examples References provided but may be limited	Provides clear descriptions for all 4 application types Lists relevant examples for each Provides adequate references	Provides comprehensive descriptions with key features for all 4 application types Lists highly relevant, diverse examples Provides high-quality references	/8
Data Transfer	Basic understanding of 1-2 concepts (data packets, IPv4, DNS) Minimal explanation of the web retrieval process	Limited understanding of all concepts Partial explanation of the web retrieval process Some inaccuracies present	Clear understanding of all concepts Mostly accurate explanation of the web retrieval process Minor details may be missing	Comprehensive understanding of all concepts Detailed, accurate explanation of the web retrieval process Demonstrates excellent grasp of how components work together	/11
Web Protocols and Ports	Completes 1-3 rows of the protocol table accurately Minimal analysis of the Dev Tools screenshot Little to no understanding of client-server model	Completes 4-6 rows of the protocol table accurately Basic analysis of the Dev Tools screenshot Limited understanding of client-server model and request-response cycle	Completes 7-8 rows of the protocol table accurately Clear analysis of the Dev Tools screenshot Good understanding of client-server model and request-response cycle	Completes all rows of the protocol table accurately Comprehensive analysis of the Dev Tools screenshot Excellent understanding of client-server model, request-response cycle, and all related concepts	/15
Securing the Web	Basic definition of 1-2 security concepts Minimal explanation of password hashing	Limited explanation of authentication and authorization Basic understanding of SSL certificates Partial explanation of password hashing	Clear explanation of authentication and authorization with examples Good understanding of SSL certificates Accurate explanation of password hashing	Comprehensive explanation of all security concepts with relevant examples Detailed understanding of SSL certificates and their importance In-depth explanation of password hashing, including salting and comparison with plain text storage	/8
Big Data	Basic definition of big data Mentions 1-2 of the three V's Minimal explanation of big data use in web applications	Limited explanation of big data and the three V's Basic examples provided Partial explanation of big data use in web applications	Clear explanation of big data and the three V's with relevant examples Good explanation of big data use in web applications with one clear example	Comprehensive explanation of big data and the three V's with highly relevant examples Detailed explanation of big data use in web applications with two clear, diverse examples	/8
THEORY TOTAL					/50

Part 2 - Progressive Web Application (100)

Task Overview

Design, develop, and implement a Progressive Web Application (PWA) for managing a collection of items. This project will demonstrate your understanding of web development concepts, database integration, and PWA features using HTML, CSS, Bootstrap, Flask, SQLite, and service workers.

Project Requirements

1. Project Setup and Planning (15 marks)

a) Project Structure (5 marks)

- Set up a Flask project with a logical folder structure
  (e.g., separating static files, templates, and Python modules).
- Initialise a Git repository and make regular commits as you develop.

b) Database Schema Design (5 marks)

- Choose one of the following themes for your collection:
  - Books
  - Movies
  - Music
  - Video Games
  - Recipes
- Design a SQLite database schema with at least five relevant columns for your collection items.
- Create a `models.py` file to define your SQLAlchemy model(s).

c) Screen Design (5 marks)

- Create wireframes or mockups for your main screens (e.g., home page, item list, add/edit item forms).
- Ensure your design is responsive and considers both desktop and mobile layouts.

Desktop 1:

Desktop 2:

Mobile 1:

Mobile 2:

2. Backend Development with Flask (20 marks)

a) Database Integration (6 marks)

- Import your database model from `models.py`.
- Use Flask-SQLAlchemy to connect to your SQLite database.
- Ensure your database operations are efficient and follow best practices.

b) Routing and Request Handling (8 marks)

- Create an `app.py` file as your main application file.
- Implement routes for all CRUD operations and additional features in `views.py`.
- Use appropriate HTTP methods (GET, POST) for different operations.

c) Query Implementation (6 marks)

- Implement at least two queries using SQLAlchemy filters (equivalent to WHERE clauses).
- Implement at least one query to sort results (equivalent to ORDER BY).
- Ensure efficient querying, especially for search and filter operations.

3. User Interface and Design (20 marks)

a) Template Structure (5 marks)

- Create a base template (`base.html`) that other templates will extend.
- Implement templates for all necessary pages (e.g., index, add item, edit item, view item).

b) Responsive Layout (5 marks)

- Use Bootstrap's grid system to create a responsive layout.
- Ensure your application is mobile-friendly and adapts to different screen sizes.

c) Bootstrap (5 marks)

- Implement at least three different Bootstrap components (e.g., navbar, cards, modals).
- Use Bootstrap's responsive features effectively.

d) Custom Styling (5 marks)

- Apply custom CSS to enhance the design without conflicting with Bootstrap styles.
- Ensure a consistent and polished look throughout the app.

4. Application Features (25 marks)

a) Create Functionality (5 marks)

- Implement a form to add new items to the collection with proper validation and error handling.

b) Read Functionality (5 marks)

- Create a page to display all items in the collection with proper formatting and pagination.

c) Update Functionality (5 marks)

- Implement the ability to edit existing items with validation and data handling.

d) Delete Functionality (5 marks)

- Add the ability to delete items from the collection with proper confirmation and error handling.

e) Search and Filter (5 marks)

- Implement a search feature to find specific items in the collection.
- Add filtering options (e.g., by year, genre, or rating).

6. Progressive Web App Features (5 marks)

- Create a manifest.json file with appropriate metadata.
- Implement a service worker (service-worker.js) for caching and offline functionality.
- Ensure the application is installable on supported devices and has basic offline functionality.

7. Version Control (5 marks)

- Make regular, meaningful commits throughout the development process.
- Use clear and descriptive commit messages.
- Create a .gitignore file to exclude unnecessary files (e.g., pycache, .env).

8. Documentation and Code Comments (5 marks)

- Include inline comments in your code to explain complex logic or functionality.
- Create a README.md file with project description, setup instructions, and usage guidelines.

9. Project Evaluation (5 marks)

- Write a brief evaluation of your project, discussing challenges faced and potential future improvements.
- Reflect on the effectiveness of your chosen design and implementation strategies.

Marking Rubric - PWA

Project Setup and Planning (15 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
Project Structure	Minimal folder structure. Flask app in a single file.	Basic separation of templates and static files. Main app logic in one file.	Well-organised structure with separate folders for templates, static files, and Python modules.	Exemplary structure with additional organisation (e.g., blueprints). Clear separation of concerns.	/5
Database Schema Design	Basic schema with fewer than 5 columns. Minimal use of SQLAlchemy.	Schema with 5 columns. Basic use of SQLAlchemy models.	Well-designed schema with appropriate data types and relationships. Effective use of SQLAlchemy.	Comprehensive schema with optimised models. Excellent use of SQLAlchemy features.	/5
Screen Design	Basic wireframes for 1-2 main screens. Limited detail.	Wireframes for most main screens. Some consideration of layout.	Clear wireframes for all main screens. Good consideration of responsive design.	Detailed wireframes or mockups for all screens. Excellent responsive design consideration.	/5

Backend Development with Flask (20 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
Database Integration	Basic connection to SQLite. Minimal CRUD operations.	Functional SQLite integration. Most CRUD operations are implemented.	Efficient SQLite integration with Flask-SQLAlchemy. All CRUD operations are working well.	Optimised database operations. Excellent use of Flask-SQLAlchemy features.	/6
Routing and Request Handling	Basic routes for home page and one or two features.	Routes for most required features. Some use of appropriate HTTP methods.	All required routes implemented correctly. Proper use of HTTP methods.	Comprehensive routing with excellent organisation. Perfect HTTP method usage.	/8
Query Implementation	Basic queries without filtering or sorting.	Implemented required WHERE and ORDER BY queries with some errors.	Correct implementation of required WHERE and ORDER BY queries.	Efficient implementation of required queries plus additional helpful queries.	/6

User Interface and Design (20 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
Template Structure	Basic templates without inheritance.	Some use of template inheritance.	Good use of base and child templates.	Excellent template structure with reusable components.	/5
Responsive Layout	Minimal responsiveness. Significant layout issues.	Mostly responsive with some layout issues.	Responsive design works well on different devices.	Perfectly responsive with optimised layouts for all devices.	/5
Bootstrap	Minimal use of Bootstrap. Only basic components used.	Basic use of Bootstrap. Some components are implemented but not fully utilised.	Good use of Bootstrap with a variety of components. Responsive grid system used effectively.	Excellent use of Bootstrap. Wide range of components used appropriately. Advanced features like custom forms or modals implemented.	/5
Custom Styling	No custom styles. Using default Bootstrap only.	Limited custom styling. Minor modifications to Bootstrap defaults.	Good custom styling. Bootstrap defaults enhanced with appropriate custom CSS.	Excellent custom styling. Thoughtful enhancements that complement Bootstrap. Consistent and polished look throughout the app.	/5

Application Features (25 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
Create Functionality	Basic form to add items with minimal validation.	Form to add items with some validation.	Fully functional form with good validation and error handling.	Excellent form with comprehensive validation and intuitive user experience.	/5
Read Functionality	Basic display of items without formatting.	Formatted display of items.	Well-formatted display with pagination.	Excellent display with sorting, pagination, and efficient data loading.	/5
Update Functionality	Basic edit form with minimal functionality.	Edit form with some validation and error handling.	Fully functional edit feature with proper validation and data handling.	Excellent edit functionality with optimised performance and user experience.	/5

Delete Functionality	Basic delete function without confirmation.	Delete function with simple confirmation.	Delete function with proper confirmation and error handling.	Excellent delete function with additional features (e.g., soft delete, undo).	/5
Search and Filter	Basic search or filter functionality.	Implemented search and basic filtering.	Effective search and filtering options.	Advanced search and filtering with excellent user interface.	/5

Progressive Web App Features (5 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
PWA Functionality	Basic manifest file. Service worker with minimal functionality.	Manifest file and service worker implemented with some offline capability.	Properly implemented manifest and service worker with good offline functionality.	Excellent PWA implementation with comprehensive offline support.	/5

Version Control (5 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
Git Usage	Few commits with unclear messages.	Regular commits with basic messages.	Consistent commits with clear messages. Good use of .gitignore.	Frequent, well-documented commits. Excellent use of Git features and .gitignore.	/5

Evaluation and Documentation (10 marks)

Criteria	Basic	Limited	Effective	Highly Effective	Mark
Project Evaluation	Minimal reflection on the project. No discussion of challenges or improvements.	Brief reflection on the project. Limited discussion of challenges faced.	Good analysis of the project, including challenges faced and potential improvements.	Comprehensive evaluation with insightful reflections on challenges, solutions, and future improvements.	/5
Documentation and Code Comments	Minimal or no code comments. Basic README with limited information.	Some code comments. README with basic setup instructions.	Good inline comments explaining complex logic. Comprehensive README with clear instructions.	Excellent documentation throughout the code. Detailed README with setup, usage guidelines, and clear explanations of project structure.	/5

PWA TOTAL	/100

Feedback: