



NSW SOFTWARE ENGINEERING 11–12 • PROGRAMMING FOR THE WEB

Week 1: Applications of Web Programming

Booklet 1 of 4 — Student Teaching Booklet

Week 1 of 30

Syllabus reference: Section 11.1 | Outcomes: SE-12-01, SE-12-03, SE-12-06

Name: _____ Class: _____

This Week at a Glance

Strand: Programming for the Web | Textbook/corpus reference: Section 11.1

Syllabus Content

- Explore the applications of web programming, including: interactive website/webpages, e-commerce, progressive web apps (PWAs).

Outcomes Addressed

Code	Outcome
SE-12-01	justifies methods used to plan, develop and engineer software solutions
SE-12-03	analyses how current hardware, software and emerging technologies influence the development of software engineering solutions
SE-12-06	justifies the selection and use of tools and resources to design, develop, manage and evaluate software

Learning Intentions

By the end of this week, you will be able to:

- Identify and describe three major categories of web application: interactive websites, e-commerce platforms and Progressive Web Apps (PWAs)
- Explain the architecture, key features and trade-offs of each application type
- Apply a decision framework (user requirements, business constraints, technical considerations) to recommend a suitable web application type for a given scenario
- Justify a technology recommendation using specific, scenario-relevant evidence

Success Criteria

- I can name the three application types and give a real-world example of each
- I can explain at least two trade-offs for each application type
- I can use the decision framework to justify a recommendation in a written response
- I can correctly answer HSC-style multiple choice questions on this topic

Contents

This Week at a Glance.....	2
Introduction: Why This Matters	4
1. Interactive Websites and Webpages.....	4
2. E-commerce Applications.....	5
3. Progressive Web Apps (PWAs).....	6
4. A Decision Framework for Choosing a Web Application Type	7
5. Worked Examples	8
6. Practice Questions	9

Introduction: Why This Matters

Web programming has grown far beyond simple static pages. Today's web is a rich ecosystem of application types, each suited to different user needs and business requirements. Every modern web application must balance user experience, performance, security, accessibility, development cost and ongoing maintenance. This week you will learn to identify three major categories of web application, understand their trade-offs, and apply a decision framework to justify which type best suits a given real-world scenario — a skill directly assessed in the HSC.

1. Interactive Websites and Webpages

Interactive websites are the foundation of modern web experiences. They provide dynamic content and real-time user interaction through forms, multimedia and client-side scripting (usually JavaScript).

Client-side / server-side architecture

A user's browser (the client) requests a page. HTML/CSS render the structure and style; JavaScript manipulates the Document Object Model (DOM) to respond immediately to user actions. Form submissions and AJAX requests send data to the server, where application logic processes the request, talks to a database, and returns a response — often without reloading the whole page.

Key features

- Dynamic content — content that changes based on user actions, time, or data
- Form processing — collecting and validating user input
- Client-side interactivity — immediate response to user actions without a full page reload
- Real-time updates — live feeds, notifications or collaborative features

Trade-offs

Aspect	Benefits	Constraints
User experience	Immediate feedback, smooth interactions	Requires JavaScript; accessibility challenges
Development	Rich client libraries, rapid prototyping	Complex state management, browser compatibility
Performance	Reduced server load, cached assets	Initial load time, network dependency
Maintenance	Clear separation of concerns	Multiple codebases (client + server)

2. E-commerce Applications

E-commerce platforms are among the most complex web applications, requiring sophisticated features for product management, payment processing and security.

Typical architecture

- Frontend: product catalog, shopping cart, checkout, user account
- Backend services: product management, order processing, payment gateway, inventory system, user management
- Data layer: product, order, user and inventory databases
- External services: payment processor, shipping API, email service

Key considerations

Security requirements

- PCI DSS compliance for payment processing
- SSL/TLS encryption for all transactions
- Secure session management
- Input validation and sanitisation

Performance challenges

- High traffic during sales events
- Database optimisation for product searches
- Image/asset delivery optimisation
- Caching strategies for product catalogues

Business logic complexity

- Inventory management and stock tracking
- Pricing rules and promotions
- Tax calculations across jurisdictions
- Shipping cost calculations

3. Progressive Web Apps (PWAs)

Progressive Web Apps combine the best features of web and mobile applications, providing a native app-like experience while remaining accessible through a normal web browser.

Core components

- PWA shell — the app's basic UI structure, loads instantly
- Service worker — a background script that intercepts network requests, enables offline functionality and background sync
- Web app manifest — a JSON file describing the app (icons, name, colours) so it can be "installed"
- Cache storage — stores assets and data for offline use

Trade-offs

Feature	Benefits	Implementation challenges
Offline functionality	Works without internet, improved UX	Complex caching strategies, data sync
Installability	Native app-like experience	Limited OS integration vs native apps
Push notifications	User engagement and retention	Browser permission requirements, battery usage
Background sync	Seamless data updates	Complex conflict resolution, storage limits

4. A Decision Framework for Choosing a Web Application Type

When choosing the appropriate web application type for a scenario, work through three groups of factors:

A. User requirements

- Connectivity — always online vs. intermittent/unreliable connectivity
- Device usage — desktop vs. mobile vs. cross-platform
- Interaction complexity — simple browsing vs. complex multi-step workflows

B. Business constraints

- Development timeline — rapid prototyping vs. long-term platform investment
- Budget — initial development cost vs. ongoing maintenance cost
- Scalability — current user base vs. projected growth

C. Technical considerations

- Performance requirements — expected response time
- Security needs — data sensitivity and compliance requirements
- Integration — existing systems and third-party services

How to use this framework in an exam answer

- State your recommended application type clearly in your first sentence.
- Select at least two factors from the framework that are most relevant to the scenario.
- Explain WHY each factor points to your chosen application type — always tie it back to a detail given in the question.

5. Worked Examples

Study how each answer below selects a web application type and justifies it using the decision framework. You will be asked to do this yourself in the Practice Questions and again in your homework.

Worked Example 1

Scenario: A local news website needs to publish community news with reader comments and a local events calendar.

Model answer

- Recommended type: Interactive website.
- Justification: The site needs dynamic content updates, reader comments and community engagement — all core strengths of an interactive website. It does not need offline access (PWA) or secure payment processing (e-commerce), so the extra complexity of those architectures is not justified here.

Worked Example 2

Scenario: An online banking service needs to let customers view balances and transfer funds securely.

Model answer

- Recommended type: A transactional platform built on e-commerce-style architecture (secure backend services + payment/transaction gateway pattern).
- Justification: The core requirement is secure, auditable transaction processing with strict authentication — the same architectural concerns (PCI-style security, encrypted sessions, backend transaction services) that define e-commerce platforms, even though no goods are being sold. This shows the decision framework is about matching architecture to requirements, not just matching to a literal label.

Worked Example 3

Scenario: Field scientists need to collect environmental data in remote locations with little or no internet connectivity, then sync it later.

Model answer

- Recommended type: Progressive Web App (PWA).
- Justification: The defining requirement is reliable offline functionality with later synchronisation — this is exactly what a PWA's service worker and background sync features are designed for, while still giving the team an installable, app-like experience on mobile devices without building separate native apps.

6. Practice Questions

Answer the following in class. Your teacher will go through solutions together — these are not for submission.

Question 1 (1 mark)

What is a key feature of progressive web apps (PWAs)?

- (A) Require a constant Internet connection
 - (B) Can be installed like native apps
 - (C) Only work in Internet Explorer
 - (D) Do not use JavaScript
-

Question 2 (1 mark)

Which language is typically used for client-side programming?

- (A) PHP
 - (B) Python
 - (C) JavaScript
 - (D) SQL
-

Question 3 (2 marks)

Explain ONE trade-off of implementing real-time features (such as live chat or notifications) in an interactive website, and describe ONE strategy to mitigate it.

Question 4 (2 marks)

Identify TWO business logic complexities that e-commerce platforms must manage which a simple interactive website does not.

Question 5 — Extended response (4 marks)

A regional hospital wants a system that lets patients book appointments online and view test results securely, and also allows doctors to review patient histories quickly during network outages at rural clinics. Recommend and justify the most appropriate web application type(s), referring to AT LEAST TWO factors from the decision framework in Section 4.

Acknowledgements

This resource references the Software Engineering 11–12 Syllabus (2022), © NSW Education Standards Authority (NESA) for and on behalf of the Crown in right of the State of New South Wales. This program is an independent teaching resource prepared for TH Software Tuition and is not affiliated with or endorsed by NESA.

Some technical explanations, worked examples and structural approaches in this program were informed by and adapted from the Software Engineering HSC Textbook (github.com/Eatham532/Software-Engineering-HSC-Textbook), an open-source educational resource used under the MIT License (Copyright 2025 Eatham532). All content here has been independently written, restructured and substantially expanded for this course. The full MIT licence text is reproduced in the Scope & Sequence document.