University of Southampton

COMP6251 Web and Cloud Applications Development Coursework

Assignment:	React Web App	**Lecturers**:	Dr Reza Rezazadeh (ra3@ecs.soton.ac.uk) Dr Eike Schneiders (eike.schneiders@soton.ac.uk)	**Weight**:	50%
Deadline:	Thursday 08/05/2025 16:01	**Feedback**:	As per ECS policy	**Effort**:	60 h per person

This coursework assesses your ability to design and build applications using a professional web development approach based on **React** and relevant **JavaScript-based frameworks**. As web development is inherently collaborative, in your professional practice as a developer, you are likely to be expected to work within a team. Thus, for this assignment you have been allocated in a pre-assigned group of **around 3 students**. In doing so, and in preparing your report and demo, you will be working towards the following learning outcomes of the module:

Subject Specific Intellectual and Research Skills
Having successfully completed this coursework you will be able to:

1. Explain the advantages of using new cloud technologies in improving web performance and scalability.
2. Evaluate client-side and server-side programming languages and frameworks.
3. Evaluate different development approaches for web and cloud applications.

Knowledge and Understanding
Having successfully completed this coursework, you will be able to demonstrate knowledge and understanding of:

4. Familiarity with alternatives front-end and back-end frameworks and platforms.
5. Techniques for testing and deploying web applications using a range of tools and cloud platforms.

Subject Specific Practical Skills
Having successfully completed this coursework you will be able to:

6. Design and implement modern web and cloud-based applications using professional tools and platforms.

Web App Requirements

A chain of fitness centres has approached your team to develop a web application for managing member registrations and improving the overall customer experience. The application should enable **members, personal trainers,** and **administrators** to interact with features like **booking personal training sessions, creating personalised workout plans, tracking training history/progress,** and **search for fitness centres belonging to the chain**. It can also include features such as trainer recommendations, in-app communication, and integration with wearable fitness devices.

Basic Features

- **New members** can register as a member by providing their names, date of birth, address, email account, and a password.
- After approval by the **administrators**, members can search for a personal trainer, send a request for training to them, and, if accepted by that trainer, book personal fitness sessions. Furthermore, they can track their progress and history of sessions, and cancel their subscription
- A **personal trainer** can receive a request from members for a personal training session in which they describe their fitness goals. They can then review, accept, reject, or suggest an alternative personal trainer based on the patient's description.
- If a personal training session takes place, the **personal trainer** can update the **training history/progress** of the member and book the next training session.
- **Members** can view their training history and view data on their training sessions, i.e., a complete history of booked sessions and total hours exercised in the last week/month. They can also view and cancel upcoming appointments or, if none are booked, schedule their next personal training session.
- When a **personal trainer** adds a training session to a **member's** profile, the **member's training history** should be updated accordingly, and a notification should be sent to the member.

No payment system should be implemented for this prototype system.

Advanced Requirements

- Integration with open data sources. You could for instance provide a map view visualisation of available fitness centres *(as an extension to the last point in bullet two above)*.
- **Customers** can sign up and log in using their social media accounts, such as Google and Facebook.
- The system will send a verification email to the customers when signing up to verify their identity.
- Deploy your Web app to the Azure cloud platform.

Implementation Notes

- Your focus should be on implementing *"Basic Features"* first and then implementing any additional functionality such as those listed under *"Advanced requirements."*
- Most marks for this assignment are allocated for your correct and efficient use of the technologies and techniques and implementation of the basic functionality.
- You should implement your app as a single page application (SPA) with a professional layout and navigation. It must provide a responsive UI to match changing screen sizes.
- You are required to implement this app using recent versions of React and your choice of backends, such as Node.js/Express/MongoDB, Firebase, or similar.

Web App demo

During the final week of the semester (Week 12*,* 12-16 May), you will be asked to demonstrate your app to the teaching team. The exact time for each group will be announced on the module website later in the semester. The purpose of this demo is to show that you have successfully implemented the required

features. A demo of your web app on your local machine (i.e., on your laptop) is considered a basic requirement, and a demo of a deployed version on Azure counts as an extra feature.

- Each group will present their system for a maximum duration of 15-minutes (live demo). Adhering to this time limit is part of the considerations during the grading.
- This will be followed by Q&A (10 minutes).

The demo and Q&A will take a maximum of 30 minutes.

Submission Instructions

Note that marks will be deducted for not adhering to these specifications. The report will be submitted at the end of week 11 (9th of May 2025).

All source files together must be submitted as a ZIP file (avoid using an alternative archive format such as RAR) to the ECS handing server by the deadline to avoid late penalties. The report must be provided in PDF format and use a font size of 11 points or larger. The report is to have a filename of the form "`team_xx.pdf`" (where "`xx`" denotes the group number identifying the team, e.g., `team_01.pdf`, `team_02.pdf`, …, `team_13.pdf`) and exactly the following contents:

Page 1: Description of prototype functionality, including an overview of the features you have implemented, listed based on each stakeholder/role. Explain any assumptions or interpretations of the requirements for this system, as they inform your approach.

Page 2: List of tools and techniques used, with a justification for your choices (meeting L.O. 1, 4). You must also provide evidence of professional software development practices, including version control. Any references, including tutorials, online code repositories and textbooks you used to support your development are to be cross-referenced as a footnote in this page. A more extended resource list may be added into the appendices for reasons of space.

Page 3: Brief overview of design and implementation, including key design decisions (L.O. 6).

Page 4: Techniques for testing and deploying web applications (L.O. 5). Give clear explanations of how the website functionalities were tested, how you tested portability, business logic, etc.

Page 5: Relevant statistics (e.g. lines of code written, plus an assessment of code taken from acknowledged external sources – provide a list giving sources). Evidence your use of a *private* GitHub repository or any similar tool supporting continuous integration (L.O. 6).

Page 6: Critical evaluation of the web application submitted. (L.O. 2, 3)

In addition to these six pages (plus a cover page indicating clearly your group number and team members), an appendix of *maximum* of 14 additional pages containing material that is cross-referenced elsewhere in your report. Examples of suitable content for the appendix section includes:

- Samples of code, using suitable formatting and colour, to highlight technologies and techniques you used, appropriately caption. You should identify which source files these samples have been extracted from.
- Any supporting design diagrams (e.g. UML diagrams, wireframes), code fragments, screenshots and other figures, each with explanatory captions (labelled Fig 1, Fig 2, …) and relevant cross references within the first 6 pages of the report. No other figures should be included.

Marking Scheme

There are three assessment criteria, each weighted as follows:

Criterion	Descript	L.O.	Weight
Demo of **Basic Features**	To what extent there was an effective implementation of required features, good use of technology (platform and frameworks), usefulness, innovation, ease of use and UX.	4, 5, 6	50%
Demo of **Advanced Features**	To what extent advanced features and techniques were designed and implemented. Other considerations such as robustness, security, and performance were included.	4, 5, 6	20%
Report	To what extent professional competence of web application development was exhibited, through a complete list of implemented functionality, clear discussion of the application design, development, and testing, supported by appropriate evidence and clear justification of design choices as well as discussion of the effectiveness of your decisions in achieving the	1, 2, 3, 4, 5, 6	30%

This marking scheme is indicative. Marks returned to students for feedback purposes are subject to moderation and before late penalties. Late submissions will be penalised as per university policy (https://www.southampton.ac.uk/~assets/doc/quality-handbook/Late%20Submission.pdf). Regulations Governing Academic Responsibility and Conduct 2024-25 apply (https://www.southampton.ac.uk/about/governance/regulations-policies/student-regulations/academic-integrity).

Make sure your documentation covers each of the main aspects of your app in sufficient detail so that your work can be assessed in line with the criteria specified above. Note that features, technologies, and techniques you have implemented but not explained in the report, or explained incorrectly, will not gain credit.

Assessment Descriptors for Guidance

The table below has descriptions indicating the attributes typically associated with each grade band.

Grade	Required Features, Technologies &	Additional Features	Report
A* (Exceptional) 80-100 marks	All required features were successfully implemented with correct and proficient use of the required technologies and techniques.	Additional features are implemented correctly and proficiently.	A succinct report with excellent structure highlighting the implemented features and providing good understanding, insight and supporting evidence. Supporting appendices showcase clear and detailed features.
A (Excellent) 70-79 marks	The required features were implemented correctly but maybe not effective in using the required technologies and techniques.	Some additional features are implemented.	A report with excellent structure highlighting implemented features and providing excellent understanding, insight and supporting evidence.

B (Very Good) 60-69 marks	Most of the required features were successfully implemented using the required technologies and techniques.	Some additional features are implemented partially.	A clear report with a good structure, highlighting implemented features and providing a very good understanding, insight and supporting evidence.
C (Good) 50-59 marks	Some required features were successfully implemented using the required technologies and techniques.	No additional features.	A well-written report with a good structure, listing implemented features and evidence of understanding and insight.
D (Weak) 35-49 marks *This is a failing grade*	A partially successful attempt to implement the required features, technologies, and techniques.	No additional features.	The report shows limited understanding, provides insufficient supporting evidence, lacks insight, and/or clarity.
F (Poor) 1-34 marks *This is a failing grade*	A weak attempt or no evidence that the required features, technologies or techniques were used correctly.	No evidence of use of advanced techniques or technologies. No consideration or evaluation.	The report is poorly written or shows little insight or understanding.