**General Requirements**

* **Project Type:** React Web Application
* **Team Size:** Pre-assigned group of around 3 students
* **Effort Required:** 60 hours per person
* **Deadline:** Thursday, 08/05/2025, 16:01
* **Demo Week:** Week 12 (12-16 May 2025)
* **Submission Format:**
  + **Source Code:** ZIP file (no RAR)
  + **Report:** PDF format (filename format: team\_xx.pdf, where xx is the group number)

**Learning Outcomes**

1. Explain advantages of cloud technologies in 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.
4. Demonstrate knowledge of alternative front-end and back-end frameworks.
5. Utilize testing and deployment techniques for web applications.
6. Design and implement modern web and cloud-based applications using professional tools.

**Web App Functional Requirements**

**Basic Features:**

* **User Registration:**
  + Members register by providing name, DOB, address, email, and password.
  + Approval by an administrator is required.
* **Trainer Search & Booking:**
  + Members can search for personal trainers and send training requests.
  + Trainers can accept, reject, or suggest alternative trainers.
* **Session Management:**
  + Trainers update training history and schedule the next session.
  + Members can view training history, total exercise hours, and upcoming sessions.
  + Members can cancel or book new sessions.
* **Notifications:**
  + Members receive notifications when a session is added to their profile.

**Constraints:**

* **No payment system implementation.**

**Advanced Features:**

* **Integration with Open Data Sources:**
  + E.g., map visualization of fitness centers.
* **Social Media Authentication:**
  + Sign-up and login via Google and Facebook.
* **Email Verification:**
  + Send verification email on sign-up.
* **Cloud Deployment:**
  + Deploy the app on **Microsoft Azure**.

**Implementation Guidelines**

* Implement **basic features** first before advanced features.
* Focus on correct and efficient **use of technologies**.
* The app must be a **Single Page Application (SPA)**.
* Provide a **responsive UI** for different screen sizes.
* Use **React** (latest version) for front-end.
* Use a backend like **Node.js/Express/MongoDB, Firebase, or similar**.

**Demo Requirements**

* **Demo Type:** Live demo in front of the teaching team.
* **Duration:**
  + **15 minutes:** Presentation and demo.
  + **10 minutes:** Q&A session.
  + **Total:** 30 minutes.
* **Deployment Considerations:**
  + Local machine demo is **basic**.
  + Azure deployment counts as an **extra feature**.

**Submission Instructions**

**Files to Submit:**

1. **ZIP File** containing all source files.
2. **Report (PDF, 11pt font minimum) containing:**
   * **Page 1:** Prototype description & stakeholder features.
   * **Page 2:** Tools and techniques with justification (incl. version control evidence).
   * **Page 3:** Design & implementation overview.
   * **Page 4:** Testing and deployment techniques.
   * **Page 5:** Relevant statistics (e.g., lines of code, external sources, GitHub usage).
   * **Page 6:** Critical evaluation of the app.

**Appendix (Max 14 Pages) for:**

* Code samples, UML diagrams, wireframes, screenshots, and other relevant documentation.

**Marking Scheme**

| **Category** | **Weight (%)** | **Evaluation Criteria** |
| --- | --- | --- |
| **Basic Features** | 50% | Correct implementation, usability, UX. |
| **Advanced Features** | 20% | Advanced techniques, security, robustness. |
| **Report** | 30% | Comprehensive documentation, software development practices. |

**Grading Criteria**

* **A\* (80-100 marks):** All features implemented excellently with strong justification in report.
* **A (70-79 marks):** Required features implemented correctly with minor issues.
* **B (60-69 marks):** Most features correctly implemented, with some partial advanced features.
* **C (50-59 marks):** Some features correctly implemented but missing key requirements.
* **D (35-49 marks):** Partial attempt at required features.
* **F (1-34 marks):** Poor or no implementation of required features.