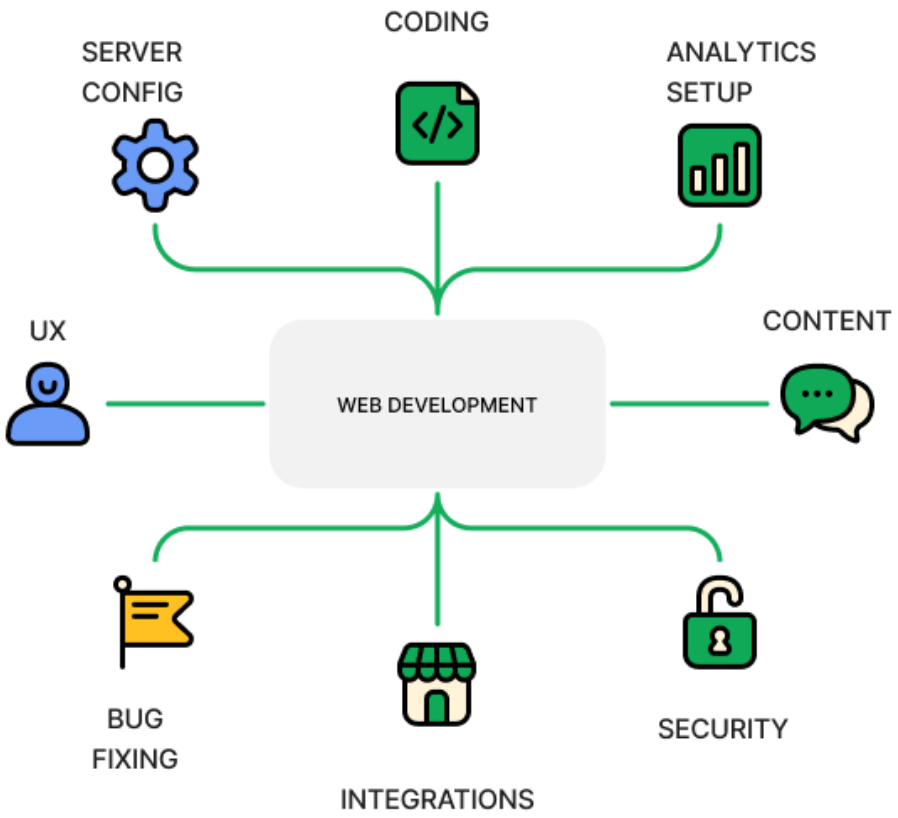
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ



Web Design and Programming

Term Project

Content

Web Design and Programming 1

Term Project 1

1. Project Overview 3
2. Project Requirements and Obligations 3
   - 2.1. Core Functional Requirements: 3
   - 2.2. Technical Architecture and Technology Stack: 3
   - 2.3. Security and Deployment: 3
3. Project Process and Timeline 4
4. Submission and Evaluation 4
5. Sample Project 4

1. Project Overview

The goal of this project is for students to build a complete, dynamic web application from scratch using modern web development technologies. The project covers the fundamental stages of web development, including database design, back-end (server-side) programming, front-end (user interface) development, and deployment in a cloud environment.

2. Project Requirements and Obligations

The items listed below are the minimum requirements that must be met for the project to be accepted.

2.1. Core Functional Requirements:

- **Database:** The project must include at least **5 database tables**. At least two of these tables must have a meaningful relationship (e.g., one-to-many, many-to-many).
- **User Management and Data Operations:** The system must include **Login** and **Register** functionalities. Additionally, the project must contain examples of all **CRUD (Create, Read, Update, Delete)** operations on the database.
- **Web Service Integration:** The project must integrate with an external web service (API) to send or receive data.
- **Interface Access:** The application must be accessible both remotely and through an embedded interface.

2.2. Technical Architecture and Technology Stack:

- **Back-end (Server-Side):**
  - **Programming Language:** Java (Recommended version: JDK 19+)
  - **Framework:** Spring Boot (Version 3.x)
- **Front-end (User Interface):**
  - **Programming Language:** JavaScript (ES6+)
  - **Embedded Page:** At least one page must be server-side rendered using **Thymeleaf**.
  - **Dynamic Pages:** The rest of the user interface must be developed using **React.js** for a modern user experience. (A Node.js environment is required).
- **Database:**
  - **System:** A **MySQL** database will be used.
- **IDE (Development Environment):**
  - IntelliJ IDEA (Community or Ultimate edition) is recommended.

2.3. Security and Deployment:

- **Project Security:** The security of the application's pages and data must be ensured. Unauthorized access must be prevented.

- **Deployment:** By the end of the term, the developed project must be deployed and running on a cloud platform. The recommended service, as demonstrated in class, is **AWS (Amazon Web Services)**.

3. Project Process and Timeline

- **Project Selection (Week 3):** Each student must determine their project topic and its details by the end of the 3rd week.
- **Development Environment Setup (Week 4):** All necessary software, libraries, and tools for the project must be installed and operational by the end of the 4th week.
- **GitHub Usage:**
  - Each student must create a private GitHub repository for the project. The repository's creation date and commit history will be checked.
  - The development process should continue throughout the semester, with code pushed to GitHub at regular intervals with meaningful commit messages.
  - The repository must have at least **5 meaningful commits** by the end of the project.
  - **Important:** Work compressed into the last few weeks will not be accepted. The development process is a significant part of the evaluation.

4. Submission and Evaluation

- **Project Submission:** The project must be submitted as a compressed **.zip file** to the project assignment on the LMS (Learning Management System).
- **Final Deadline: Tuesday, December 16th, at 23:00**. Submissions after this date and time will not be accepted under any circumstances.
- **Additional Note:** The course instructor reserves the right to make additions to the project document or requirements if deemed necessary during the project period.

Expected Deliverables:

At the end of the term, each student is expected to deliver the following:

1. A working back-end application.
2. A persistent data layer (database).
3. A user-friendly front-end interface with all functionalities working.
4. Basic documentation summarizing the project's setup and architecture.
5. A link to the project deployed and accessible on a cloud platform (e.g., AWS).

---

5. Sample Project

- **Student Name:** John Smith
- **Project Topic:** Online Library Management System

Brief Description:
A system where users can search for books, borrow, return, and reserve them. It will include an admin panel for managing books and members. Book information (such as cover image, author, publisher, etc.) will be fetched from an external book API.

Sample Database Tables:

1. `users` (Users)
2. `books` (Books)
3. `loans` (Lending Transactions) - related to the `users` and `books` tables.
4. `reservations` (Reservations) - related to the `users` and `books` tables.
5. `categories` (Book Categories)