Proposal Date:\_\_Aug 25, 2024\_\_\_\_\_\_\_\_\_

**AIT CASE STUDY COVER SHEET**

Course

 AIT 710

X AIT 715

 AIT 720

 AIT 725

 AIT 730

 AIT 735

Student Name: Lei Peng

Semester: Fall 2024

Case Study Title: WeLearn Database Website Project

Advisor Name:

|  |
| --- |
| **Abstract**  WeLearn is an overseas education agency that assists Chinese high school students in applying to American colleges. Recently, the company established a MySQL database as its information management system, which is currently hosted on a local computer serving as both the client and server, accessible only by staff.  The objective of this case study is to design a user-friendly website that enables both clients and staff to access the database via the Internet. The project involves developing a website with JavaScript as the front-end technology, using Java and Spring Boot for the backend, and deploying the server on AWS EC2. This solution aims to enhance accessibility and improve the efficiency of the company’s operations. |

Past Courses: Please list all course completed and currently taking. Remember that you must complete all prerequisites in order to register for a case study.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Number*** | ***Name*** | ***Semester*** | ***Grade*** |
| AIT500 | Fundamental Computer Programming/Data Structure | Fall 2020 | S |
| AIT 600 | Information Technology Infrastructure | Fall 2020 | B |
| AIT 610 | Systems Development Process | Fall 2020 | A- |
| AIT 616 | Fundamental Web Tech/Development | Spring 2021 | B+ |
| AIT 632 | Database Management | Spring 2021 | A |
| AIT 612 | Information System Vulnerability/Risk Analysis | Spring 2021 | B+ |
| AIT 618 | Client/Server-Side Programing Web | Fall 2021 | B |
| AIT 624 | Software Engineering Fundamental | Fall 2021 | B+ |
| AIT732 | Advanced Database Management System | Fall 2021 | F |
| AIT628 | Information Technology and Business Strategy | Spring 2022 | A |
| AIT642 | Software Testing & Management | Spring 2022 | A- |
| AIT732 | Advanced Database Management System | Spring 2022 | A- |
| AIT630 | Information Technology Project Management | Fall 2022 | A |
| COSC 501 | Fundamental Data Structures and Algorithm | Fall 2022 | A |
| AIT614 | Network Security | Spring 2023 | F |
| AIT620 | Business Data Communications | Spring 2023 | C |
| COSC 502 | Computer Organization Assembly Language | Spring 2023 | U |
| AIT614 | Network Security | Fall 2023 | A |
| AIT622 | Network Architecture/ Protocols | Spring 2024 | A- |
| AIT735 | Case Study: Database Management | Spring 2024 | A |
| AIT665 | Cloud Computing | Spring 2024 | A |

**FOR OFFICE USE ONLY**

Proposal Approved:

Report Received:

Case Study: WeLearn Database Website

# Problem Description

WeLearn is an overseas education agency that helps Chinese high school students apply to American colleges. Despite having fewer than 10 employees, the company serves clients across China. To streamline operations and improve efficiency, WeLearn has established a centralized MySQL database early this year. Currently, the database is only accessible to staff and is hosted on a local computer that functions as both server and client. With the successful implementation of a pilot module, it's now time to develop a user-friendly website that allows both staff and clients 24/7 access to the database from anywhere in the world.

**Project Scope**

This project aims to develop a user-friendly web application that enables WeLearn staff and clients to access the WeLearn database over the Internet. The application will streamline customer relationship management processes, improve data accessibility for employees, and enhance communication with students.

**Scope Description:**  
The web application will provide the following functionalities:

1. **User Authentication and Authorization:**

* Implement a secure login system for employees.
* Use JWT (JSON Web Tokens) for authentication and authorization
* For further security all the passwords in the database would be hashed to improve security
* Role-based access control, ensuring only authorized personnel can view and modify specific data.

1. **Customer Relationship Database Management:**

* Allow employees and clients to access, add, update, and delete student records in the CRM database.
* Include fields for storing essential student information (e.g., contact details, course and university information).
* Use different triggers and stored procedures in database level as well to ensure that the data inserted is consistent and reliable.
* ERD diagrams for our database will also be generated using workbench.

1. **Reporting and Analytics:**

* Provide basic reporting tools to generate insights on student information and application status.
* Allow data export in common formats (e.g.PDF).

1. **User Interface and Experience:**

* Develop an intuitive, responsive web interface that is accessible on both desktop and mobile devices.
* Ensure ease of navigation for users with different levels of technical expertise.

1. **Data Security and Compliance:**

* Implement encryption for sensitive data in transit and at rest.
* Ensure the application complies with relevant data protection regulations.
* Ensures that it follows all the OWASP principles for a secure system.

1. **Scalability:**

* Design the application to accommodate the potential growth of the company in terms of the number of students and employees.
* Follow SOLID and DRY principles while coding so the code is easier to understand and is manageable.

**Project Deliverables:**

* Requirements Documentation
* Database ER diagram
* User interface diagram
* Website source code
* User manual
* Database creation SQL scripts
* List of software used to build and run the project

# Justification of the problem (2 -3 paragraphs)

**Topical Alignment**

This case will serve as a real-world project for the company WeLearn. It involves the development of a database-driven website. The design and implementation of this project can be applied to many similar businesses.

**Learning elements**

**1. Front-End Development**

* **HTML/CSS:** Understanding the basics of web structure and styling using HTML (HyperText Markup Language) and CSS (Cascading Style Sheets).
* **JavaScript:** Learning how to make web pages with this widely used open-source general-purpose scripting language.
* **Responsive Design:** Use frameworks/libraries like React.js to build dynamic, responsive user interfaces.
* **Front-End Frameworks:** Familiarity with popular frameworks Vue.js to build complex user interfaces.

**2. Back-End Development**

* **Server-Side Programming:** Use Express for building the server-side logic.
* **Database Management:** Understanding how to design, create, and manage databases using MySQL databases.
* **APIs and Web Services:** Creating RESTful service in Java for communication between the front-end and back-end.
* **Authentication and Authorization:** Implementing user authentication and managing roles and permissions within the application.

**3. Full-Stack Development**

* **Integration of Front-End and Back-End:** Learning how to connect front-end interfaces with back-end logic and databases to create a cohesive application.
* **Version Control:** Using GitHub for version control and collaboration.

**4. DevOps and Deployment**

* **Server Configuration:** Setting up and configuring web server Apache to host the web application.
* **Continuous Integration/Continuous Deployment (CI/CD):** Understanding automated testing, building, and deployment processes using tools like GitHub.
* **Cloud Platforms:** Familiarity with cloud services like AWS EC2 for scalable hosting and deployment.

**5. User Experience (UX) Design**

* **User-Centered Design:** Learning the principles of designing applications with the end-user in mind.

**6. Project Management**

* **Agile Methodology:** Understanding Agile principles and practices, including Scrum and Kanban, for managing development projects.
* **Requirement Gathering:** Learning how to work with stakeholders to define application requirements and translate them into technical specifications.
* **Time and Resource Management:** Estimating the time and resources needed for project completion and managing them effectively.

Each of these elements contributes to a comprehensive understanding of web application development, making it possible to build, deploy, and maintain effective and secure web applications.

**Complexity and Size**:

The project encompasses various aspects of website development and database management, offering both learning challenges and manageable scope within a graduate capstone timeframe.

# Preliminary Design Plan

This design plan aims to create a cohesive system where all components work together harmoniously. The scalability of the architecture will be considered to allow for future growth, including the potential increase in the number of students and employees. Furthermore, the design will include provisions for integration with existing tools and systems, as well as potential expansions such as advanced analytics or mobile app development in later phases.

**Step 1: Establish a MySQL database**

The database has already been established but can be further improved during the course of the project. Please refer to attached ER Diagram and the Create Tables files.

**Step 2: Backend development**

* Setup the backend using NodeJS and ExpressJS.
* Install necessary packages: ExpressJS, JWT, mysql2, dotenv, Sequelize.
* Setup the routes for agents, students, schools, etc. Implement authorization and protection for admin and client routes.
* Test the routes in Postman.

**Step 3: Front End development**

* Install the application using ReactJS.
* Setup the routes and protection (for admin and clients).
* Connect Frontend with Backend and communicate using https to get and send the data.
* Display the data.
* Make the website responsive.

**Step 4: AWS Deployment:**

* Link the GitHub project with AWS.
* Install the necessary dependencies on the AWS server and deploy the backend.
* Create a MySQL instance on AWS and connect it with our backend.
* Install necessary dependencies and deploy the front end of the application.
* Change the environmental variables on the front end so that they can communicate with the deployed backend using https protocol.

**Major Use Cases**:

* User log in and log out of the site with user account.
* To use the database per user account role.

For Employees: Users will be able to have administrative authority to use the database:

* To create, read, update, and delete (CRUD) data of their own students’ student profiles, applications, and contracts.
* To CRUD data of all the other information such as College, Programs, Schools, exams information.
* To conduct calculation of the contract amount.
* To conduct inquires based on filters about their own students’ information, application status, college, programs information.
* To track all of their students’ application process.

For Clients: Users will be able to use the student account to grant the access for their own accounts for the following:

* To read and update their own student profile information.
* To read information from tables of College, Programs, and Application Status
* Track their own application process.

* To provide customized inquire tools to search college information, client information, applications status).

**Sitemap**

**A diagram of a website

Description automatically generated**

# Preliminary Technology Framework

|  |  |  |
| --- | --- | --- |
| **Technology** | **Purpose** | **Description** |
| ExpressJS | To make routes and APIs | The framework of NodeJS |
| NodeJS | To connect to the database | A backend coding language |
| ReactJS(JavaScript) | Provide dynamic interactivity on websites | A programming language for HTML document |
| JSON | To transfer data from backend to frontend | JavaScript Object Notation for data-interchange format |
| HTML | To create Web page | A markup language for structuring and presenting content on the web |
| CSS | To control the design of the webpage | A style sheet language to present and style of a document written by HTML |
| MySQL | To establish the database | A relationship database |
| GitHub | For codes version control | A code repository |
| Mac OS X | To support all the application programs | The operating System |
| Visual Studio Code | To write and edit codes | A source code editor |
| AWS EC2 Instance | To deploy the database to AWS EC2 | Cloud computing |
| AWS Elastic Load Balancer | To distribute incoming traffic across multiple EC2 instances | AWS load balancer |
| Microsoft Word | Documentation | A documents editor |
| AWS S3 | To store unstructured data | Cloud object storage |
| CloudFront | To store user uploaded images | A content delivery network (CDN) service |
| Elasticsearch | To search text in the database | a distributed, RESTful search and analytics engine |
| MySQL2 | To communicate between NodeJS and MySQL | MySQL client for Node.js with focus on performance. |
| Balsamiq | To draw user interface diagram | A Wireframing tool |
| JWT | To securely transmit information between client and server | A URL-safe means generating tokens for authorization |
| Sequelize | For relationships with different tables. | A modern TypeScript and Node.js ORM for MySQL |
| dotenv | To store the credentials like Database name, Database password, database user, PORT, JWT | A Centralized configuration management module |

**Preliminary Architecture:**

**Back End**

EC2 Instance

JavaScript, Express

MySQL Database

AWS RDS/EC2

**Front End**

JavaScript-React.js

Load Balancer

AWS ELB

* + - * **Top Level:** The user interface (frontend) communicates with the backend via HTTP/HTTPS.
      * **Middle Layer:** A load balancer receives incoming requests and distributes them to one or more EC2 instances running the Express application.
* **Bottom Layer:** The backend application server communicates with the MySQL database (hosted on RDS or EC2).

# Preliminary Schedule

WeLearn Project is implemented in following steps.

|  |  |  |
| --- | --- | --- |
| **Activities** | **Estimated Time (in hours)** | **Meeting Date** |
| 1. **Requirement Gathering and Analysis)**  * Gather user and stakeholder requirements * Analyze business needs and define the scope of the application. * Setup development environment (For the backend, using NodeJS and ExpressJS, For the frontend, ReactJS) * Finalize with user manual, user interface diagram | 10 hours  (2 week)  Aug 26 – Sep 6 | Sep 7, 2024 |
| 1. Design & Architecture  * Design system architecture and components * Create wireframes and UI/UX design for frontend * Define API specifications and endpoints * Current database schema design and ERD creation review & improvement * Set up AWS infrastructure and networking (VPC, subnets, security groups) | 20 hours  (2 weeks)  Sep 9- Sep 20 | Sep 23, 2024 |
| 1. Development   **Week 1**   * Set up development environment and CI/CD pipeline * Begin backend development (ExpressJS setup, basic services) * Start frontend development (JavaScript framework setup)   **Week 2 (Oct 7 - Oct 13):**   * Continue backend development (APIs, service layers) * Develop core frontend components and layout   **Week 3 (Oct 14 - Oct 20):**   * Implement database integration (MySQL setup, CRUD operations) * Build out additional frontend features and components   **Week 4 (Oct 21 - Oct 27):**   * Integrate frontend with backend APIs * Develop user authentication and authorization   **Week 5 (Oct 28 - Nov 3):**   * Implement advanced backend features (business logic, error handling) * Finalize frontend functionalities (forms, validation, UI/UX improvements)   **Week 6 (Nov 4 - Nov 10):**   * Perform initial integration testing * Fix integration issues and refine code | 60 hours  6 weeks  Sep 23-Nov 1 | Oct 7  Oct 21  Nov 4 |
| 4: Testing  **Week 1 (Nov 11 - Nov 17):**   * Conduct functional and unit testing * Start performance testing and load testing * Log and prioritize bugs and issues   **Week 2 (Nov 18 - Nov 24):**   * Perform end-to-end testing * Conduct user acceptance testing (UAT) with stakeholders * Fix remaining bugs and issues * Finalize testing and prepare for deployment | 30 hours  3 weeks  Nov 4 – Nov 22 | Nov 25 |
| **5: Deployment Preparation and Launch**  **Week 1 (Nov 25 - Dec 1):**   * + - * Prepare production environment on AWS EC2 and RDS       * Set up security configurations and backups       * Conduct final reviews and approvals   **Week 2 (Dec 2 - Dec 7):**   * + - * Deploy application to production environment       * Perform smoke testing on production       * Monitor application performance and resolve immediate issues       * Official launch of the application | 20 hours  Nov 25 – Dec 6 | Dec 8, 2024 |
| **6: Deliver & Presentation** | 10 hours | Dec 13, 2024 |
| Total | 150 hours |  |

# Reading list

T. Connelly, and C. Begg. *Database Systems: A Practical Approach to Design, Implementation, and Management*” 6th Edition. 2010.

I. Sommerville. *Software Engineering* 9th Edition. Addison-Wesley, 2010.

J. Duckett. Javascript & JQuery: Interactive Front-end Web Development. Wiley. 2014.

J. Duckett. HTML & CSS: Design and Build Websites. Wiley. 2014.

I, \_\_\_\_\_Lei Peng\_\_\_\_\_ propose to complete this project during the \_\_Fall\_\_ semester of \_\_*2024*\_ and understand that this project and its derived materials (e.g., source code, written reports, presentation slides) are to reflect my own work, unless explicitly and appropriately referenced. Furthermore, I understand that plagiarism or other unattributed use of material not written by me is completely unacceptable and **will be considered sufficient cause for a failing grade on the project.** For additional information on academic integrity policy at Towson University, I will visit [www.towson.edu/provost/resources/studentacademic.asp](http://www.towson.edu/provost/resources/studentacademic.asp) .

Student’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructor’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Graduate Program Director’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_