KIET Group of Institutions, Ghaziabad *CSE Department*

Internship Report

on

Summer Internship at MLSA KIET

Internship name: Full-Stack Web Development

(2024-25)

Submitted By: Tarun Kumar Sangal
Branch: CSE(Sec-5C)
Roll No: 2200290100171

ACKNOWLEDGEMENT

I’ve got this golden opportunity to express my kind gratitude and sincere thanks to my Head of Institution, KIET Group of Institutions of Engineering and Technology, and the Head of the Department of “**Computer Science and Engineering”** for their kind support and necessary counselling in the preparation of this project report. I’m also indebted to every person responsible for the making up of this project directly or indirectly.

I must also acknowledge our deep debt of gratitude to each one of my colleagues who led this project to come out the way it is. It’s my hard work and untiring sincere efforts and cooperation to bring out the project work. Last but not least, I would like to thank my parents for their sound counselling and cheerful support. They have always inspired us and kept our spirits up.

Name : Tarun Kumar Sangal
Branch: CSE
Semester: V
University Roll No: 2200290100171

Introduction

The Full Stack Web Development Internship organized by the Microsoft Learn Student Ambassadors (MLSA) and the Computer Science and Engineering Department at KIET Group of Institutions provided me with hands-on experience in building full-stack web applications. Throughout the internship, I gained practical skills in both front-end and back-end technologies, fostering my ability to develop user-centered, responsive web applications. This internship emphasized learning industry-relevant tools and methodologies, enabling a robust understanding of full-stack web development.

Objectives

The primary objectives of the internship included:

- Building technical expertise in web development.
- Enhancing problem-solving skills applicable to real-world scenarios.
- Preparing for competitive opportunities in the tech field.
- Developing collaboration skills through Git and GitHub for version control.
- Gaining a deep understanding of modern web development technologies.

Phases of the Internship

The internship was structured into two major phases, each with specific learning goals:

Phase 1: Learning Modules and Microsoft Badges

This phase consisted of **40 hours of structured learning** through Microsoft Learn. The topics covered included:

- **Core Concepts of Full Stack Web Development**: HTML, CSS, JavaScript, and React framework for front-end design and interactivity.
- **Server-Side Development**: Node.js for implementing server-side logic.
- **Database Management**: MongoDB for data storage and retrieval.
- **Version Control**: Git and GitHub for repository management and collaboration.

Upon completion of these modules, I earned **Microsoft Badges** which showcase my achievements and proficiency in these areas.

Phase 2: Project Work and Repository Management

The second phase consisted of **60 hours of hands-on project work** where I applied the concepts learned in Phase 1. I chose to complete one easy project: **Create a JavaScript quiz** and one Intermediate Project: **Create a simple calculator**

Key areas of focus in this phase were:

- **Project Planning and Execution**: Working on projects with defined objectives, categorized as Easy, Intermediate, and Difficult.
- **Repository Management**: Utilizing Git for version control and making pull requests (PRs) to manage changes effectively.
- **Code Reviews and Collaboration**: Collaborating with peers and resolving queries to refine and optimize project work.

Project Summary

During the internship, I completed the following projects:

Project 1:

Title: Create a Javascript Quiz (Difficulty Level:Easy)

Objective

The objective of the JavaScript quiz application is to create an interactive, user-friendly platform that allows users to test their knowledge on various topics by answering multiple-choice questions. The application provides instant feedback on answers, keeps track of the score, and offers a summary at the end, enhancing user engagement and learning.

Technologies Used

1. **HTML**: Structuring the layout and elements of the quiz, including the question display area, answer buttons, and score tracker.
2. **CSS**: Styling the quiz interface for a clean, visually appealing user experience, including responsive design adjustments for various screen sizes.
3. **JavaScript**: Adding interactivity, handling question logic, user responses, scoring, and managing data flow between questions.

Key Features

1. **Dynamic Question Loading**: Questions are dynamically loaded from a predefined list, allowing for easy modification and addition of new questions without changing core logic.
2. **Real-Time Feedback**: Instantly informs users if their answer is correct or incorrect, and updates the score in real-time.
3. **Score Tracking**: Tracks user score and displays it at the end of the quiz to provide feedback on their performance.
4. **Timer and Progress Indicator** (optional advanced feature): Adds a countdown timer and/or progress bar, motivating users to answer within a certain time and providing a sense of quiz progress.

Challenges Overcome

1. **Data Flow and State Management**: Managing the flow of questions, user answers, and scores in a way that maintains the quiz state across multiple questions. This required careful tracking of the current question index and score, ensuring smooth transitions between questions.
2. **Dynamic DOM Manipulation**: Effectively using JavaScript to manipulate the DOM, updating question text, choices, and other UI elements without reloading the page.
3. **Error Handling and Validation**: Ensuring the application doesn’t break if there are unexpected inputs or errors in question data, including graceful handling of cases where a question might lack answers or have invalid data.
4. **User Experience Design**: Balancing clear feedback, intuitive design, and appropriate spacing between elements to ensure the interface remains uncluttered and easy to navigate.

Project 2:

Title: Create a Simple Calculator Difficulty Level:Intermediate)

Objective

The objective of this calculator application is to create a functional, user-friendly tool that can perform basic arithmetic operations like addition, subtraction, multiplication, and division. The calculator provides a clean, responsive design, making it easy for users to input numbers and view results instantly.

Technologies Used

1. HTML: Structures the layout of the calculator, including the display area and buttons for numbers and operations.
2. CSS: Styles the calculator interface, ensuring a clean, visually appealing layout with appropriate spacing and button sizes.
3. JavaScript: Handles all the arithmetic logic, user interactions, and updates to the display, making the calculator interactive.

Key Features

1. Responsive Design: Ensures that the calculator works on various devices and screen sizes, including mobile and desktop.
2. Basic Arithmetic Operations: Supports addition, subtraction, multiplication, and division.
3. Clear and Delete Functionality: Includes options to clear the entire display or delete the last entered character for error correction.
4. Real-Time Calculation Display: Updates the display in real-time with each input, providing instant feedback for users as they type.
5. Keyboard Support (optional feature): Allows users to input numbers and operations through the keyboard, improving accessibility.

Challenges Overcome

1. Display Management: Ensuring the display updates correctly for each input, without any unintended concatenations or display issues. This required careful handling of the display string to format it appropriately and clear it when needed.

2. Operation Handling: Implementing logic to handle various operations without errors, especially when switching between different operations (like going from addition to multiplication).

3. Error Handling: Accounting for edge cases, such as division by zero or pressing multiple operators in a row, to ensure the calculator doesn’t crash or show incorrect results.

4. User Experience Optimization: Designing a layout with button sizes and spacing that’s intuitive and easy to use, especially on smaller screens, to improve usability and readability.

Skills Acquired

Through this internship, I developed the following skills:

- **Technical Skills**: Full-stack web development, responsive design, and server-side programming.

- **Version Control and Collaboration**: Proficiency in Git and GitHub, working with pull requests.

- **Problem Solving**: Improved logical thinking and debugging skills through practical projects.

Conclusion

The Full Stack Web Development Internship at KIET was an enriching experience that equipped me with essential skills for a career in technology. The exposure to industry-relevant tools and methodologies allowed me to build strong technical and collaboration skills that will be valuable in future projects and professional settings.