

# **Address Updater Project Assessment Report**

## **Introduction:**

The Address Updater project is a web-based application built using HTML, CSS, JavaScript, JSP (JavaServer Pages), and MySQL. The project is hosted on the Tomcat server, and Eclipse is employed as the integrated development environment. It focuses on providing a user-friendly interface for managing addresses, with data stored in a MySQL database.

## **Strengths:**

### **1. Intuitive UI:**

- The user interface, designed with HTML and CSS, ensures easy navigation and a clean visual layout.
- Utilization of JavaScript enhances the interactivity and responsiveness of the application.

### **2. Real-Time Updates:**

- The use of JSP allows for dynamic content generation, enabling real-time updates in the user interface.

### **3. Error Handling:**

- Effective error handling, facilitated by JavaScript, ensures data integrity by preventing duplicate entries and enforcing constraints on fields.

### **4. Database Choice:**

- MySQL is chosen as the database platform, providing reliability, scalability, and ease of integration.

### **5. Separate Update and Delete Buttons:**

- HTML and JavaScript are leveraged to implement distinct buttons for updating and deleting addresses, contributing to a streamlined user experience.

### **6. Server Platform:**

- Tomcat server is used for hosting the application, offering a robust and scalable environment for Java-based web applications.

### **7. Development Environment:**

- Eclipse serves as the integrated development environment, providing tools and features to streamline the development process.

## **Weaknesses:**

### 1. Deletion Confirmation:

- Lack of a confirmation step during address deletion may lead to accidental data loss.

### 2. Security Concerns:

- Absence of user authentication poses security risks, and addressing this is crucial for data protection.

### 3. City and State Verification:

- Users can input random city and state information without validation.

### 4. Zip Code Validation:

- Zip codes are not verified against their associated city or state, impacting data accuracy.

## **Recommendations:**

### 1. Google Maps Integration:

- Explore options to integrate Google Maps using JavaScript for users to pinpoint their exact location.
- Store this location along with address details for enhanced accuracy.

### 2. User Authentication:

- Implement user authentication using JSP for a secure login and logout functionality.
- Store user data separately in the MySQL database to ensure data privacy.

### 3. User Interface Styling:

- Enhance the visual appeal of the user interface with improved HTML and CSS styling.
- Apply design elements to make the application more attractive and user-friendly.

### 4. Search and Filtering:

- Implement search functionality using JavaScript for quick address retrieval.
- Add filters (e.g., by city, state, zip code) to enhance data accessibility.

### 5. User Roles and Permissions:

- Define different user roles using JSP and MySQL with varying access levels (e.g., admin, regular user).
- Assign appropriate permissions to each role for data security and integrity.

**Conclusion:**

The Address Updater project demonstrates strengths in its technology stack, hosting platform, and development environment. However, addressing the identified weaknesses and implementing the recommended enhancements will result in a more secure, feature-rich, and visually appealing application for managing addresses.