Course ID: DLBCSPJWD01 **| Student:** Mohana Ramakrishnan **Matriculation Number:** 92129139 **Tutor:** Christian Remfert

Project Overview

The Banking Web Application project aims to provide a secure, efficient, and user-friendly platform for individuals and small businesses to manage their financial activities, and secondary users, such as bank administrators, oversee transactions, manage user accounts, and ensure system security.

Benefits of the application

This Banking application offers User Authentication- Registration, login, and multi-factor authentication, Account Management- View balance and transaction history, Fund Transfers- Single and recurring transactions and Budgeting Tools- Expense tracking and budgeting assistance benefiting users with efficient and effective transactions.

Software technologies

The application will be built with React JS as front end, Express.js as back end and PostgreSQL as Database.

The following are the technical components of the system.

- **Frontend: ReactJS** was chosen for its component-based architecture, enabling efficient UI development and management.
- **State Management: React's** built-in **us estate and useEffect hooks** manage state, with **Context API or Redux** for complex state needs.
- **Styling: CSS modules or styled-components** were used for component-level styling, ensuring modular and maintainable code.
- **Backend: ExpressJS** handles server-side logic, paired with **PostgreSQL** for robust, relational data management.
- **Authentication: JWT or Passport.js** ensures secure user authentication and session management, critical for banking operations.

System Architecture

- The architecture of this project consists of a Client Side developed with React JS for managing the user interface.

- **The Server Side uses Node JS and Express JS** to handle business logic and API requests.
- **HTTP Requests** from the client are processed by the server, which interacts with the **PostgreSQL Database**.
- **HTTP** Responses are sent back from the server to the client, updating the browser's **UI Layer.**
- The **PostgreSQL Database serves** as the data layer, storing and retrieving data as needed by the server.

Takeaway from the Project

- **Database Change:** Switched from **MongoDB to PostgreSQL** for better relational data management and transaction handling.
- **Styling Framework:** Transitioned from **Bootstrap/Tailwind CSS to CSS modules** and styled-components for more modular, component-specific styling.
- **State Management:** Initially planned to rely on basic state management but considered **Context API or Redux** for complex state needs.
- **Technical constraints:** Adjustments were done to improve design flexibility, and got feedback from initial user testing.
- **Impact:** These changes aimed to enhance performance, maintainability, and user experience while aligning better with project requirements.

Conclusion

The Banking Web Application enables users to manage their finances online efficiently. It solves the problem of needing in-person banking by offering digital customer management and transaction processing. The app targets individuals and businesses seeking convenient and accessible financial services. By streamlining banking processes, it reduces time and effort, enhancing the overall user experience. The application adds value by providing an intuitive interface with secure, seamless transactions for users.