Medicare Application Capstone Project

This document contains sections for:

- Sprint planning and Task completion.
- Core Concepts used in the project.
- Technologies and Tools Used in the Project.
- The flow of the Application.
- Demonstrating the product capabilities, appearance, and user interactions.
- Unique Selling Points of the Application.
- Conclusions.

➢ The code for this project is hosted at
https://github.com/Dinesh123527/Simpli_Phase_5_Capstone_Project_Medicare-Front-End

➢ This project is developed by **G V Narasimha Raju.**

Sprints planning and Task completion:

The project is planned to be completed in 3 sprints. Tasks assumed to be completed in the sprint are:

1st Sprint:

- Creating the flow of the application.
- Initializing the git repository to track changes as development progresses.
- Creating a Spring boot application to fulfill user requirements.
- Configure the Database to maintain the data used for the application.
- Adding the Required dependencies used for the application.

2nd Sprint:

- Implemented the Model Layer and Service Layer for the project.
- Implemented the Business Logic part.
- Creating API methods to fetch the response.
- configure the API to check the Response of the requested object.
- Testing in the Post-Man Tool to check whether the API methods are working or not by Response.

3rd Sprint:

- Creating an Angular application to fulfill user requirements.
- Adding the Required Packages used for the application.
- Creating Components and Services, Styling the Pages.
- Creating Reactive Forms and Validations to the Forms.
- Creating Service methods to fetch the response.
- Passing the Response from the Service to the component class.
- Implemented the Query params using routing.
- Integrated the Payment Gateway using Stripe API for payments.

Core concepts used in the project:

• Object-Oriented: used to create and model objects for users and their credentials.

• Reactive Forms: Reactive forms provide a model-driven approach to handling form inputs whose values change over time along with Validations.

• Component: Angular components are a subset of directives, always associated with a template.

• Lifecycle hooks: ngOnInit()

• Data Binding: Data binding deals with how to bind your data from component to HTML DOM elements (Templates). We can easily interact with applications without worrying about how to insert our data. We can make connections in two different ways "1-way" and "2-way binding".

• Directives: In Angular, Directives are defined as classes that can add new behavior to the elements in the template or modify existing behavior.

• Routing: The process of defining the navigation element and t associated view is called routing in Angular. Angular provides a separate module, the Router module, for setting up navigation in an Angular application.

• Services: Angular Services is a piece of reusable code with a focused purpose. A code that you will use across multiple components in your application.

• Databases: used to store and retrieve data.

• Data Sources: used to define a set of properties required to identify and access the database.

• Collections: used some collections such as array-list to store collection of data.

• Collections: used Java8 Streams to filter and fetch a collection of data.

• Custom Exception Handling: used to catch problems that arise in the code, especially in I/O blocks.

• MVC: Micro Service is an architecture that allows the developers to develop and deploy services independently.

• Micro Services (Spring Boot): Micro Service is an architecture that allows developers to develop and deploy services independently.

• RXJS Concepts: RxJS provides an implementation of the Observable type, which is needed until the type becomes part of the language. The library also provides utility functions for creating and working with observables.

Behavior Subject.

• API: API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.

Technologies and Tools Used in the Project

- **Angular:** To create Quiz Application.

- **Angular Material:** Angular Material is a User Interface (UI) component library that developers can use in their Angular projects to speed up the development of elegant and consistent user interfaces.

- **Visual Studio Code:** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle.

- **Spring Boot:** To create an E-commerce project for Sporty Shoes.

- **Postman:** To the API request and response.

- **MySQL:** To create and manage the database.

- **JPA:** To manage the operations for the application.

- **Lombok:** It is the tool of the java library that was used to generate code for minimizing the unused code.

- **Rest-API:** To create the API methods and to check the response of the object entities.

- **Spring Tool Suite:** To write and execute the code.

- **Tomcat:** To deploy application.

Flow of the Application:
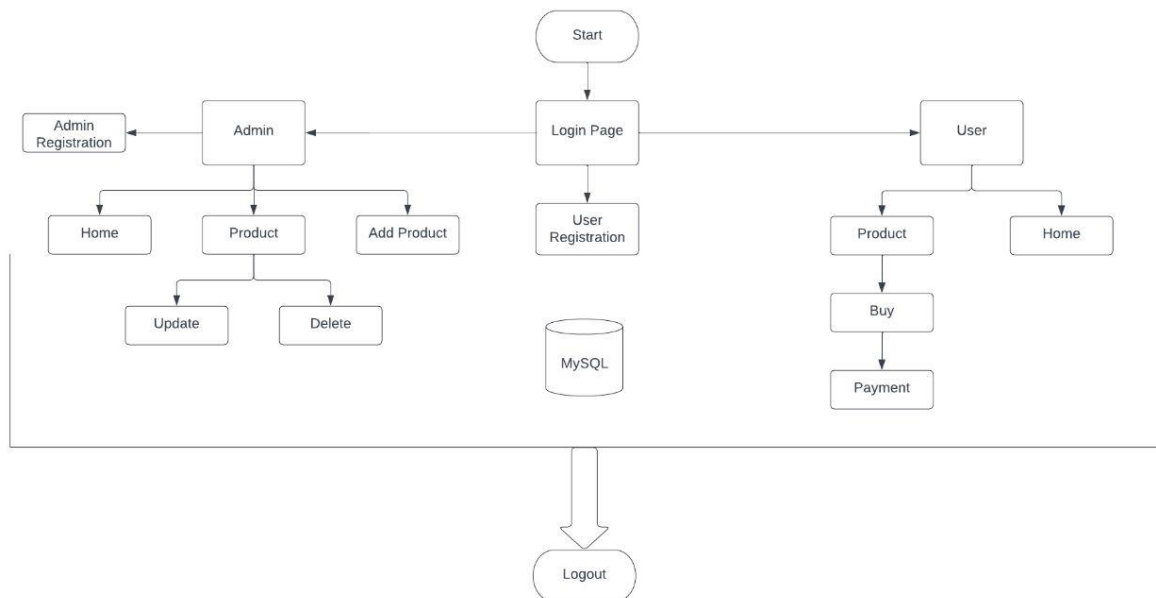
Demonstrating the product capabilities, appearance, and user interactions:

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project.

- ✓ **Creating the project using Spring Initialize.**
- ✓ **Add the required dependencies and generate the project from Spring website.**
- ✓ **Open Spring Tool Suite Open the Project from the directory and build it.**
- ✓ **Add the required Maven Dependencies in Pom.XML file and build the maven.**
- ✓ **Configure the properties in the application.properties file and run it as spring boot application.**
- ✓ **After that create new Angular project using ng new project name.**

- ✓ **Add the required packages to the project via node package manager npm command.**
- ✓ **Open the Angular Project in that go to package.json to view the added dependencies.**
- ✓ **Now Run the code using ng serve command and the application will be deployed in the default port of localhost:4200.**
- ✓ **Pushing the code to the GitHub repository.**

Unique Selling Points of the Application:

- ✓ This was Full Stack Capstone Application developed using Angular Frame work as Front End and Spring Boot as Backend where user can login from Database and User can also be Registered.

- ✓ The Application has One login Page where it acts as both Customer and Admin Dashboard.

- ✓ Implemented all the Function related both customer and admin customer can see available Medicines and can purchase them, but whereas only admin can perform the crud operations on products.

- ✓ Also Implemented the Payment gateway using Stripe and Rich UI Flow for all the Components along with Alert services.

Conclusions:

Further enhancements to the application can be made which may include:

- ▪ Need to Create the Product Image dynamically getting from associated with each project instead of displaying image data statically.

- ▪ Need to Add Login using Mobile number OTP and provide the Authentication using Google services.
- ▪ Need to display the message out of stock if any product is not available dynamically.

- Can Implement the Real time Payment Authentication Scenario instead of using 3rd Party API.