Medicare

This document contains the following:

- Project and developer details
- Sprint planning and tasks achieved
- Core concepts used in the project
- Flowchart of the application
- Links to the GitHub repository
- Demonstration of product capabilities, appearance, and user interactions
- Unique Selling Points of the application

Description

Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

Background of the problem statement:

Medicare is a company that supplies medicines and a couple of other healthcare essentials at an affordable price. It was established in 2012 in Delhi, India. It had been serving fine all these years, however, the business analysts noticed a decline in sales since 2017. They found out that online ordering of medicines with companies, such as 100mg and mfine are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop a healthcare web application with a rich and user-friendly interface.
You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

Features of the application:

1. Registration
2. Login
3. Payment gateway
4. Searching
5. Filtering
6. Sorting
7. Dynamic data
8. Responsive and compatible with different devices

Recommended technologies:

1. Database management: MySQL and Oracle
2. Backend logic: Java programming, NodeJS
3. Frontend development: JSP, Angular, Bootstrap, HTML/CSS, and Javascript
4. Automation and testing technologies: Selenium, Jasmine (frontend testing), and TestNG
5. DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS

Project development guidelines:

- The project will be delivered within four sprints with every sprint delivering a minimal viable product.
- It is mandatory to perform proper sprint planning with user stories to develop all the components of the project.
- The learner can use any technology from the above-mentioned technologies for different layers of the project.
- The web application should be responsive and should fetch or send data dynamically without hardcoded values.
- The learner must maintain the version of the application over GitHub and every new change should be sent to the repository.
- The learner must implement a CI/CD pipeline using Jenkins.
- The learner should also deploy and host the application on an AWS EC2 instance.
- The learner should also implement automation testing before the application enters the CI/CD pipeline.
- The learner should use Git branching to do basic automation testing of the application in it separately.
- The learner should make a rich frontend of the application, which is user-friendly and easy for the user to navigate through the application.

- There will be two portals in the application, namely admin and user portal.

Admin Portal:
The admin portal deals with all the backend data generation and product information. The admin user should be able to:

- Add or remove medicine details from the application to build a rich product line
- Edit medicine details like name, price, seller, product description, and offers to keep the product information updated with the current prices
- Enable or disable a medicine product

User Portal:
It deals with the user activities. The end-user should be able to:

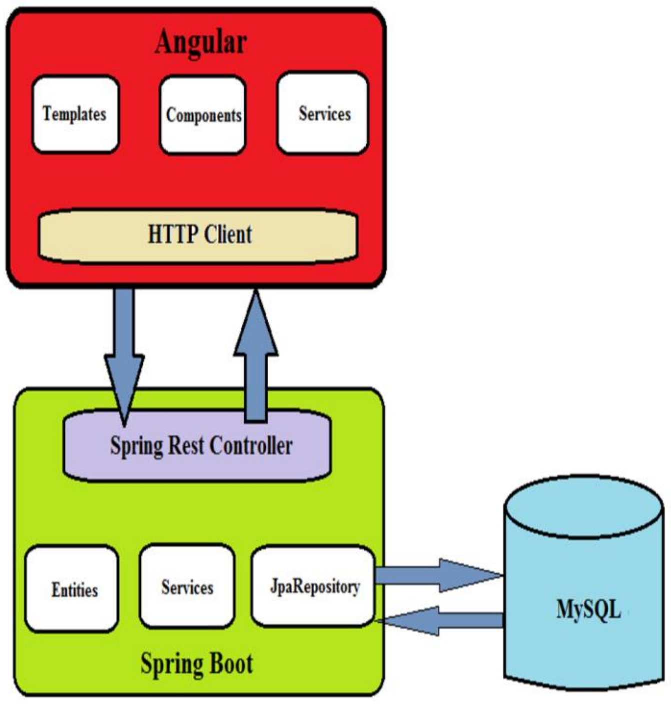
- Sign-in to the application to maintain a record of activities
- Search for products based on the search keyword
- Apply filters and sort results based on different cuisines to get the best deals
- Add all the selected food items to the cart and customize the purchase at the end
- Perform a seamless payment gateway
- Get an order summary details page once the payment is complete

Product Features:

1. Medicare application is made specifically to the required business needs. It is completely flexible and scalable to the business demands and growth.
2. The whole application is a Single Page Application that is more efficient in terms of processing and provides a seamless user experience.
3. The application web pages are responsive and secure.
4. The application has one Administrator. The Admin Portal features are:
    - The admin can login with username and password in admin portal.
    - The admin can add or remove medicine details.
    - Edit medicine details like name, price etc., to keep the product information updated with current prices. Enable or disable the medicines.

5. The application has a User portal. The User Portal features are: Sign up and login with username and password.
    - The User can maintain the record of activities.
    - Search for products based on the search keyword.
    - Apply filters and sort result based on different categories.
    - Add the product to cart and customize the purchase at the end.
    - Experience a seamless payment experience.
    - Receive an order summary once the payment is successful.

Core Concepts Used and Project Architecture:



• Angular framework for frontend UI's.

• Spring boot framework for backend .

• MySQL Database for storing all the data.

• HTML, Bootstrap 4. • Typescript.

- Spring Security and JWT Authentication.
- Spring Data Jpa, Spring Web

Developer Details:

Mahendra Kumar Singh

mahendrakumarsingh9893@gmail.com

Links to the GitHub repository:

https://github.com/Mahendra1272/Medicare.git

Pushing the code to GitHub Repository:

- Open Git Bash and navigate to the folder where you have created your files.
  cd Medicare \Spring Tool Suite\Medicare
- Initialize the repository using the below command **git init**
- Add all the files to your git repository using the below command **git add .**
- Commit the changes using the below command **git commit -m "Initial commit"**
- Add the URL for the remote repository where your local repository will be pushed **git remote add origin https://github.com/Mahendra1272/medicare .git**
- Push the files to the folder you initially created using below command **git push -u origin master**

Sprint Planning and Task Achieved:

Number of sprint planned = 4.

Sprint 1:

1. Planned to develop backend code for project. Generated Spring boot project from http://start.spring.io.
2. Planned to develop the rest api's to create Admin and User. Used spring security and Jwt authentication to achieve this task.
3. Planned to develop api's for admin portal to add, update, delete, enable or disable products.
4. Successfully developed and tested the admin portal rest api's using Postman software.
5. Planned to develop frontend code for project. Generated Angular project using angular cli.
6. Planned to develop login ui for admin and user portal. Successfully developed the ui's for admin and user.
7. Planned to develop admin dashboard that enables admin to perform the required functionalities. Successfully developed the admin dashboard.

Sprint 2:

1. Planned to develop home page of the application. Successfully developed the home ui of the application.
2. Planned to develop Sign up ui for users. Developed successfully.
3. Planned to develop ui for search product based on keyword, show product based on category. Successfully developed the ui's for user portal feature.
4. Planned to develop user home ui. Developed successfully.

Sprint 3:

1. Planned to develop add to cart feature ui in user portal. Developed successfully.
2. Planned to develop rest api's to create an order and to view orders by user. Successfully developed and tested the user order api's using postman software.
3. Planned to develop create a new order ui in user portal. Developed successfully.
4. Planned to develop ui's for order summary, show all orders in user and admin portal. Developed successfully.

Sprint 4:

1. Planned to test the complete web application by giving the required inputs in respective fields.
2. Successfully tested all the admin portal features and user portal features.
3. The Web application is responsive, secure and all features are working as per the given requirements.