MEDICARE CAPSTONE PROJECT

Prepared by: Shubham Mehta

Project Objective :

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

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

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.

Accomplishments Of The Application :

1. Smart AI Disease Detection
2. Easy to use Calorie Calculator
3. Exercise tracking
4. Search for Doctors near you
5. Consume less time and work effectively
6. Custom Scroll bar

Project 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.

Setup Guide :

- Dataset used
- Diabetes: Pima Indian Diabetes Dataset
- Heart: Heart Disease Dataset
- Kidney: Chronic Kidney Disease Dataset
- Breast Cancer: Winconsin Breast Cancer Dataset
- Remote backend URL
- (https://medicare-backend.herokuapp.com/)
- Prerequisites
- Required to install and run the software:
- npm
- Installing and Running
- From the project folder, run these commands in console (terminal) to install dependencies and run the app:
- npm install
- npm run start
- How to Contribute
- Take a look at the existing Issues or create a new issue!
- Fork the Repo, create a branch for any issue that you are working on and commit your work.
- Create a Pull Request (PR), which will be promptly reviewed and given suggestions for improvements by the community.

Setup Guide :

- Add screenshots or screen captures to your Pull Request to help us understand the effects of the changes that are included in your commits.
- HOW TO MAKE A PULL REQUEST
- 1. Start by making a fork the medi-Care repository. Click on the symbol at the top right corner.
- 2. Clone your new fork of the repository:
- git clone https://github.com/<your-github-username>/medi-Care
- 3. Set upstream command:
- git remote add upstream https://github.com/mohit200008/medi-Care.git
- 4. Check the remotes for this repository.
- git remote -v
- 5. Navigate to the new project directory:
- cd medi-Care
- 6. Create a new branch:
- git checkout -b <YourBranchName>
- 7. Sync your fork or local repository with the origin repository:
- In your forked repository click on "Fetch upstream"
- Click "Fetch and merge".

Setup Guide :

- Alternatively, Git CLI way to Sync forked repository with origin repository
- git fetch upstream
- git merge upstream/main
- Github Docs for Syncing
- 8. Make your changes to the source code.
- 9. Stage your changes and commit:
- git add .
- git commit -m "<your_commit_message>"
- 10. Push your local commits to the remote repository:
- git push origin <YourBranchName>
- 11. Create a Pull Request!