**Medicare**

Course-end Project 1

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

Or

**MyMoviePlan**

Course-end Project 2

DESCRIPTION

Create a dynamic and responsive web application for booking movie tickets online for different genres and languages.

**Background of the problem statement:**  
NMS Cinemas is a chain of single screen theatres that screen movie shows of different genres and languages at very genuine prices. It was established in 2004 in Pune, India. Recently, the business analysts noticed a decline in sales since 2010. They found out that the online booking of movie tickets from apps, such as BookMyShow and Paytm were gaining more profit by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop an online movie ticket booking 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

Front end : React JS

Backend : Spring boot with data

Database : MySQL

Two actor or type of users

Admin : SignIn no Singup for Admin account

We can check admin username and password from database

Or in spring boto service layer but no signup for admin account.

Customer : SignIn as well as SignUp for customer

In Spring boot application

Login bean class, emailId, password, typeofuser

Setter and getter methods

LoginController

LoginService

LoginRepository

Database Level

Login table (emailed(PK), password , typeOfUser)

[admin@gmail.com](mailto:admin@gmail.com), admin@123, admin

[ravi@gmail.com](mailto:ravi@gmail.com), raj@123, customer/user

[steven@gmail.com](mailto:steven@gmail.com), steven@123, customer/user

in React JS

need to install axios, react-dom modules.

App.js

Login.js

Customer.js

Admin.js

After check admin emailed ad password from database using axios service connecting

To spring boot project

If admin login open Admin Home Page

If customer login open Customer Home Page.

Admin.Js

1. Add Medicine details or Add Movies
2. Remove medicine details or remove Movies
3. Update medicine details or update movies
4. View all medicine details or View all movies

Medicine

Medicine table

Mid(PK), MName, price, qty, medicine image, company name, description etc

Customer :

Display all medicine details in images format with button as place order.

Customer can search medicine details base upon name, category, price etc

Customer can filter base upon any property like category, dieses etc.

Customer can sort a to z name or lower to high ie prize

Whenever we create customer details then please create dummy account in database

As

Accno, emailed(pk), initial amount ($1000), cardNumber

In Spring boot

Medicine Java bean

Medicine Controller

Medicine service

Medicine repository

Movies

1. Add Medicine details or Add Movies
2. Remove medicine details or remove Movies
3. Update medicine details or update movies
4. View all medicine details or View all movies

Medicine

Movies table

Mid(PK), MName, price, qty, movies image, actor details, typeOfMovies etc.

In Spring boot

Movies Java bean

Movies Controller

Movies service

Movies repository

Login

Emailed (pk)

Password

Typeofuser

Medicine or Movies

Mid, mname, price, qty, description, imageurl

Mid, mname,price, numberOfTicketAvl, moviesdecription, castdetails, imageurl

Orders or MoviesBookingDetails

Oid, emailed, mid, totalamount etc

BookId, emailed, movieid, numberOfTicket, totalamount etc

OrderProductRelationship

Opid mid, oid , qty etc

Account

Accno(Pk), emailed(fk), amount etc

**create two folder with name as backend and fronted**

open the spring initializer

