**Full Stack Java Developer**

Capstone Project Problem Statement



**Drive Your Way**

**Problem statement:**

Design and develop an online car selling and buying portal.

**Scenario:**

**Drive Your Way Ltd.** is a company working in the business of selling and buying old cars. However, due to the pandemic and lockdown, their business took a hit. They were not able to achieve the decided targets. So, they have decided to go online to increase the revenue.

**Expected deliverables:**

**Features of the application:**

1. Home Page
2. Login Page
3. Register Page
4. Subscription Plans and Pricing Page
5. Car Categories Page
6. Shortlisted Products Page
7. Admin Page
8. Adding filters in the search option

**Recommended technologies:**

1. Database management: MySQL
2. Backend logic: Java Programming (Spring Boot, JPA, Hibernate)
3. Front-end development: Angular, HTML/CSS, and Bootstrap
4. Automation and testing technologies: Selenium and TestNG
5. DevOps and production technologies: Git, GitHub, Docker, and Jenkins
6. Optional implementation: 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 use the concept of continuous development and integration for version control.
* The learner should also deploy and host the application on an AWS EC2 instance.
* The learner should make a rich front-end of the application, which is user-friendly and easy for the user to navigate through the application.
* The learner should implement validation within the backend Spring Boot layer, which will ensure data passed by the API meets the constraints of the system.
* The learner should host the backend application on a local server, and the hosted API should be used by Angular to communicate with the backend.

**Functionalities of the admin user:**

* Login
* Registration
* Adding, updating, and deleting categories, and car details which include name, description, price, location, offers, and owner details
* Adding, updating, and deleting user details

**Functionalities of end users:**

* Any member can register and view cars available for selling and can also add details of their own car for selling
* Cars and categories listing
* Shortlisting the cars the user is interested to purchase to the cart
* Full and quick view of each listed car’s information
* User can add/edit/delete car details from user’s own post/shortlisted cars
* Only people with subscriptions can view the seller details
* Includes a form where a prospective buyer can indicate interest in purchasing a car

**Frontend Validation:**

* Valid format for email, phone number and password (one uppercase, one lowercase, one special character, one number, and a minimum of 8 characters overall) during user registration.
* All necessary details are entered when listing a car for sale on the system, and in valid format.

**Backend Validation:**

* When a car is listed for sale, the registration number of the vehicle does not match an existing listed car, to ensure no duplicate listing.
* No duplicate registration of users.
* Validating each user’s privileges before allowing edit or view access to specific details. For example, only registered users can view seller details. Also, only the user who created the post for a car can edit it.

**Custom Exceptions for Business Logic:**

* Address Producer-Consumer problem with multithreading. For example, if a user wants to book a car at the same time the listing is being deleted.
* A user can fill the form indicating interest for a particular car only once, to avoid duplication/spam.