VELOCITY VEHICLES

VELOCITY VEHICLES

Speed into savings with your perfect car

PRESENTED BY:
Byte Warriors

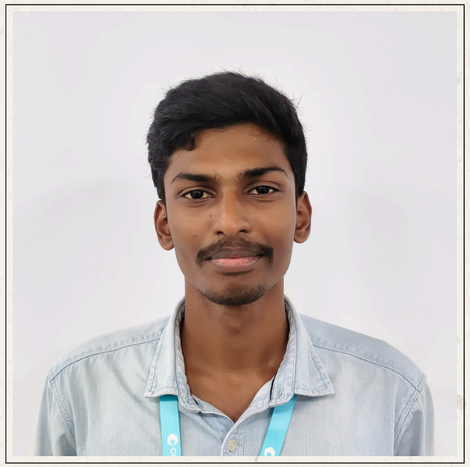
Meet our team

Aniket Pal
Team Lead

Java , python spring boot, angular, Deep learning, selenium

Aditya Raj

Java, spring, springboot, ReactJS, MySQL



Pierrs I K

java, Python , C, MySQL, sprinboot, ML, LLM’s

Project Overview

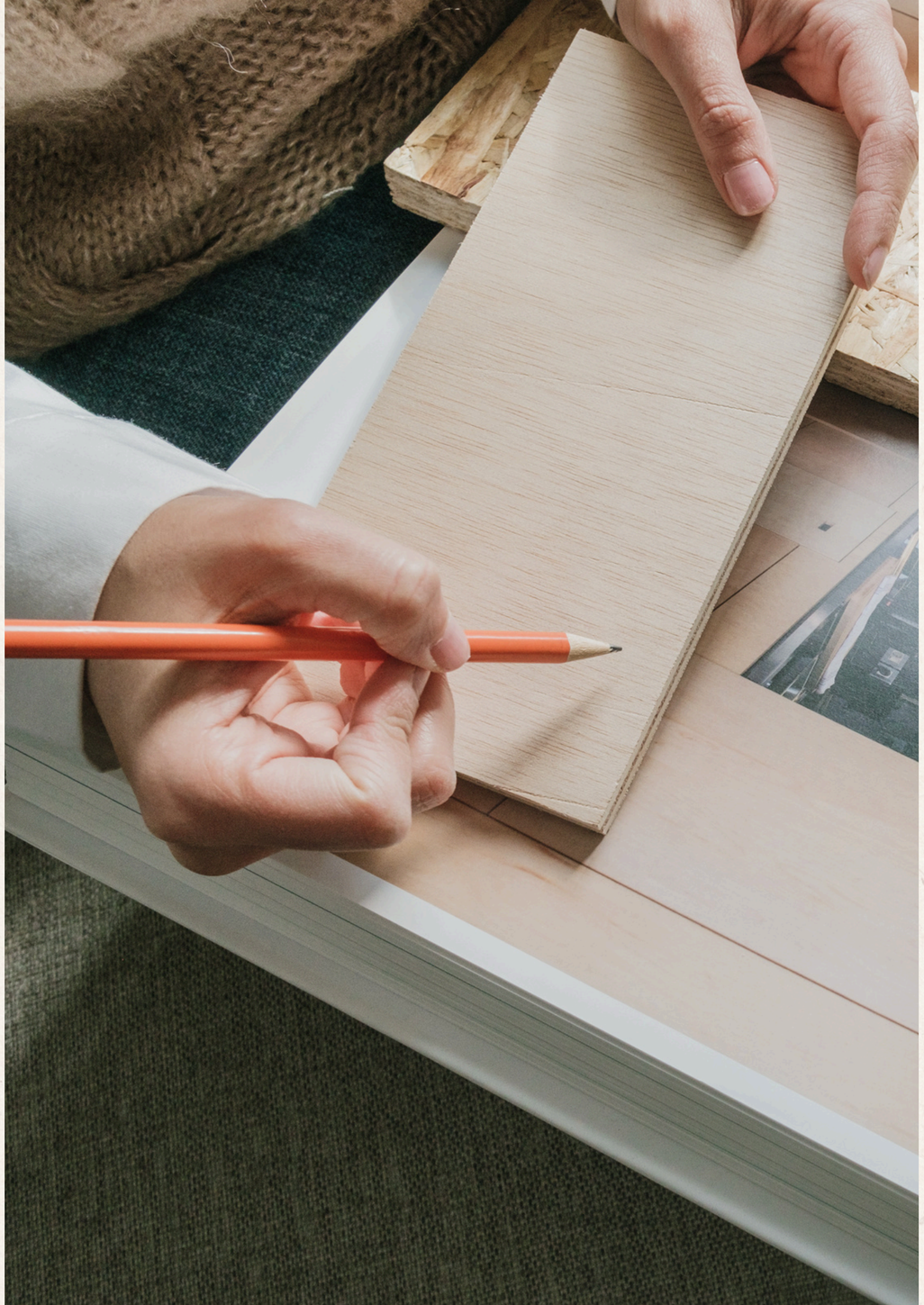
Velocity Vehicles is a web-based application designed for an auto shop, facilitating car exploration, comparison, and management for both customers and dealers. The platform aims to provide a seamless and efficient experience with a secure backend, intuitive UI, and responsive design.

Byte Warriors

01 The Functional Requirements include an interactive home page with navigation links and filters, detailed sections for exploring cars, and distinct user roles for customers and dealers.

02 The Technical Design uses MySQL, Java Spring Boot, and Angular. It features a well-structured database and a layered backend architecture for scalability and maintainability.

03 The User Interface is user-friendly and responsive, with a scrollable home page, detailed car specifications, and easy sign-up and login processes for users and dealers.



Project Initiation

The project was initiated by our trainer, providing us with a simple problem statement that aimed to let us explore java programming concepts such as interfaces, abstract and collections.

The project was then extended for the final assessment, for which we followed proper product development lifecycle.

FRD

LINK TO FRD

TDD

LINK TO TDD

Functionalities

These are the list of functionalities that are avalible to the end user in our application.

- Customer Login
- Customer Register
- Buy Car
- Book Test Drive
- View Car Details
- Search Cars
- View Profile

- Dealer Login
- Dealer Register
- View Cars
- Add Cars
- View Sold Cars
- View Sales
- Dashboard

Filter by

- price
- mileage
- fuel
- transmission
- brand
- color & more

Exception Handling

In our application we have multiple custom exceptions that try to cover any and all failures and edge cases.

To handle all those exceptions that might arise and provide the right response we have the GlobalExceptionHandler class that provides a single point of contact to all the exceptions that might arise.

This helps code readability, understanding and allows us to add and update exceptions much more easily in the future.

List of Custom Exceptions

1. CarNotFoundException
2. CarBrandNotFoundException
3. CustomerNotFoundException
4. DealerNotFoundException
5. DuplicateEmailException
6. DuplicatePhoneException
7. DuplicateUsernameException
8. InvalidCarStatusException
9. InvalidSalesDataException
10. SalesNotFoundException

Security

The security for our project was implemented with spring security. We have done so due to spring security's ease of integration within the project.

1. **Configuring Security**: Setting up security configurations to protect URLs and endpoints, using SecurityFilterChain.
2. **Authentication**: Implementing form based login and JWT (JSON Web Token) for user authentication.

Database Authentication

1. **User Details Service**: A custom UserDetailsService implementation is created to load user-specific data from the database.
2. **Password Encoding**: Passwords are securely hashed using a PasswordEncoder (BCrypt).
3. **Authentication Manager**: Configured to use the custom UserDetailsService and PasswordEncoder for authentication.

JWT token implementation

1. **JWTUtil**: A utility class created to generate and validate JWT tokens.
2. **Token Generation**: Upon successful login, a JWT token is generated and returned to the client.
3. **Token Validation**: The JWT token is validated to ensure the user is authenticated and authorized to access the resource.

Frontend

The frontend of our project is developed with Angular to deliver an efficient, component-based single-page application.

Component Based architecture

Every part of the application is built as an component so that we can easily and effiently navigate errors and increase code maintainablility

Services and DTO’s

The application has various services, each used to contact and communicate with various api endpoints. The DTO’s provide a way for us to store, transfer and serve data in our applicarion

Auth

The application authenticates users to distinguish them as either a CUSTOMER or DEALER.

Thank you

BYTE WARRIORS

Aniket Pal

Aditya Raj

Pierrs I K