VEGE Smart Parking System – Project Execution Document

1. Project Overview (Contains Hardware connections, Software, with its Integration)

- **Project Name**: VEGE (Versatile Entry and Guidance Enhancer)
- **Objective**: Develop a smart parking system using Flutter, Firebase, ESP32-CAM, and IR sensors to manage:
  - License Plate Recognition (LPR)
  - Slot Reservation
  - Indoor Navigation
  - Time-based Billing – Automatic Payment
  - User Interfaces

2. Project Execution Steps

Phase 1: Planning and Requirements Gathering

Step 1.1: Define Requirements

- User types: Registered User
- Features: Entry using LPR, slot reservation, wallet system, billing, indoor navigation
- Hardware: ESP32-CAM, Servo Motor, NodeMCU, IR Sensors
- Backend: Firebase (Auth, Firestore, Realtime DB, Storage, Functions)

Step 1.2: Design Architecture

- **System Components**:
  - ESP32-CAM → LPR
  - NodeMCU + IR Sensors → Slot Detection
  - Firebase → Realtime DB
  - Flutter App → Frontend

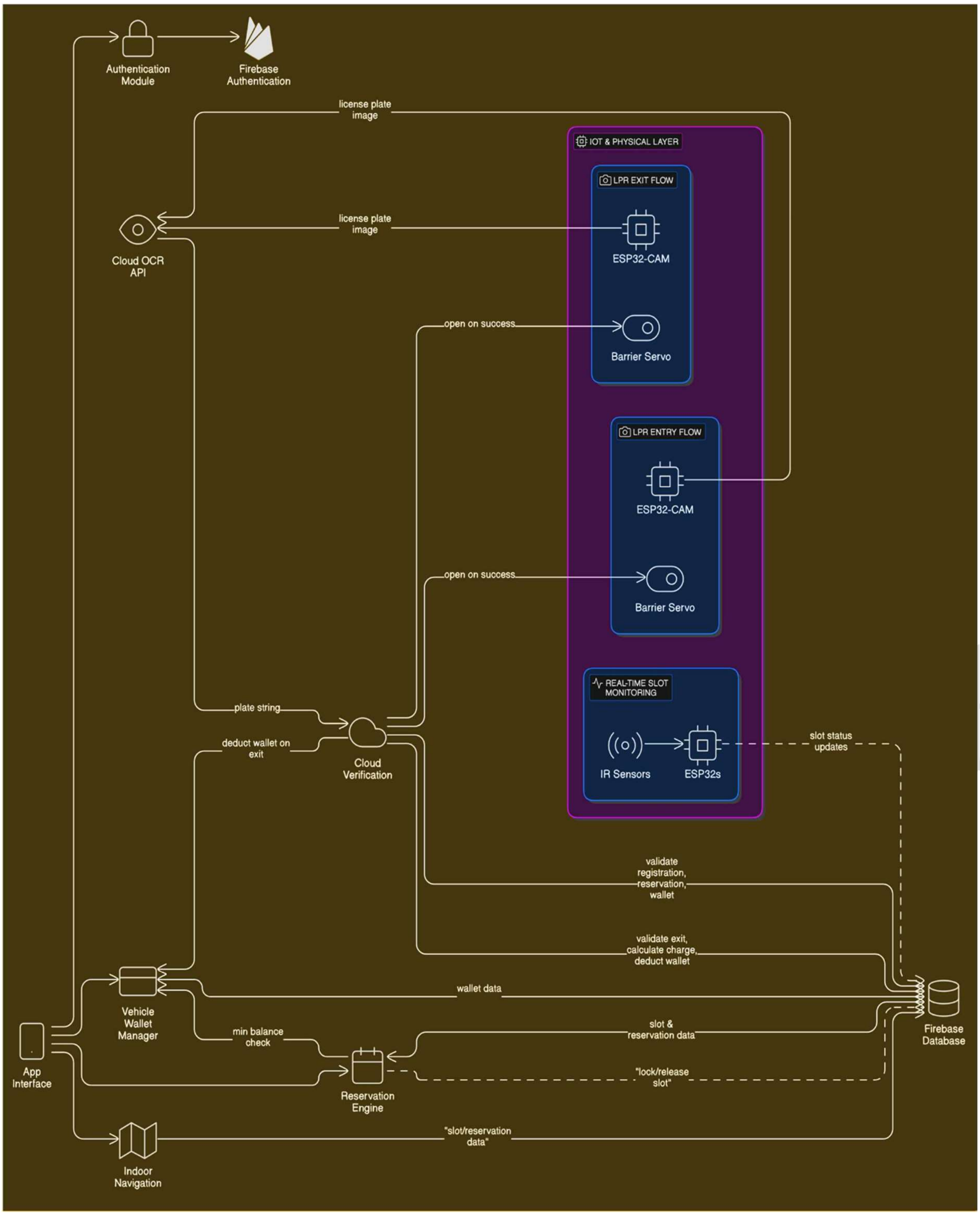


Fig 1. Component level architecture of VEGE

Phase 2: Hardware Setup and Integration

Step 2.1: ESP32-CAM Integration

- Connect ESP32-CAM to WiFi
- Capture image when vehicle arrives
- Send image to server/API for LPR
- Use Firebase to store plate number + timestamp

Step 2.2: Servo Motor + Gate Control

- Control servo using recognized plate number
- Open gate if number is valid and matches Firebase record

Step 2.3: NodeMCU Slot Detection

- Use IR sensors to detect occupancy
- Red (occupied), Green (available), Grey (reserved)
- Upload slot status to Firebase Realtime DB

Step 2.4: Display

- Show detected number and entry result (Allowed/Denied)

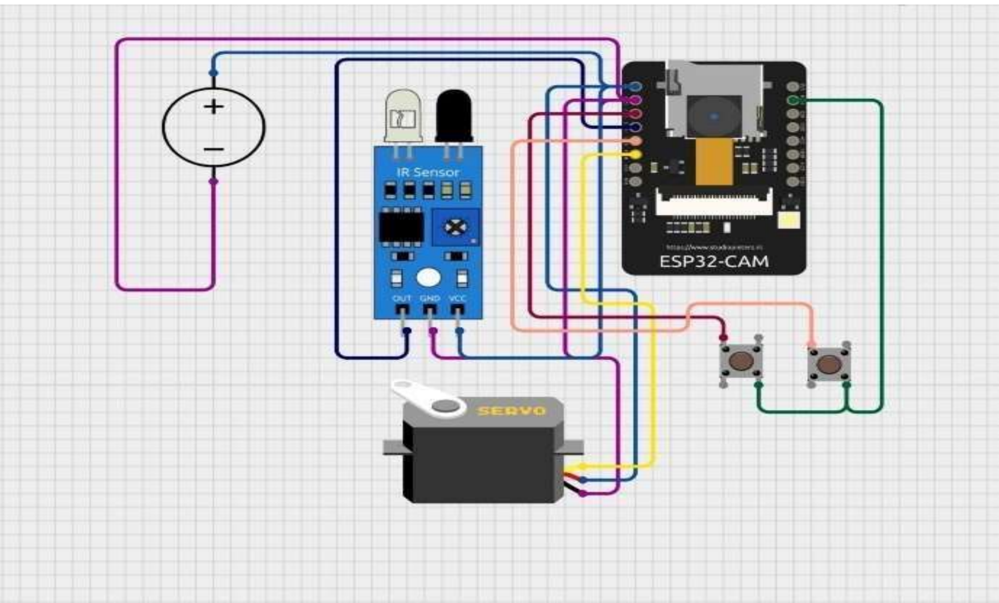


Fig 5. ESP32 module connection with IR sensor for Entry and Exit of the vehicle

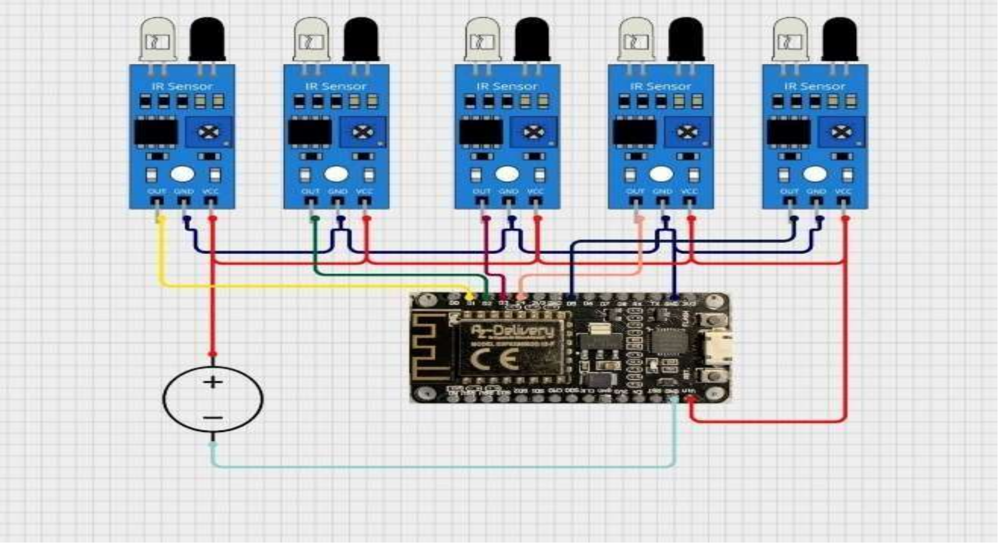
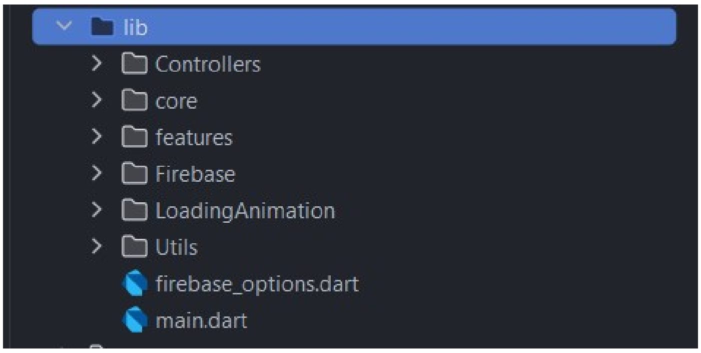


Fig 6. IR sensors connection with ESP32 for Slot Update

Phase 3: Software Development

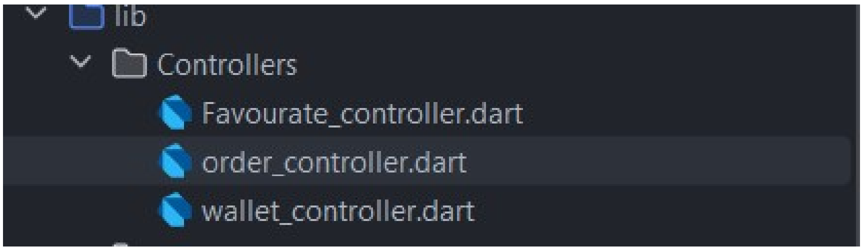
REPO WALKTHROUGH

Library Folder


```
lib
  Controllers
  core
  features
  Firebase
  LoadingAnimation
  Utils
  firebase_options.dart
  main.dart
```

- This folder contains the main coding components for the entire application

Controllers Folder



```
lib
  Controllers
    Favourate_controller.dart
    order_controller.dart
    wallet_controller.dart
```

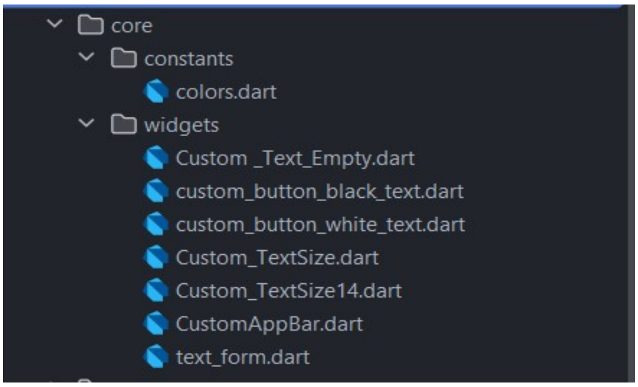
- The Controllers Folder Having 3 Dart Files

  1. Favourate_controller.dart file : Manages the functionalities for the favourite button.

  2. Order_controller.dart file : this file contains the functionalities code to manage the Orders.

  3. wallet_controller.dart file : these file contains the functionalities code to manage the Wallet.

Core Folder

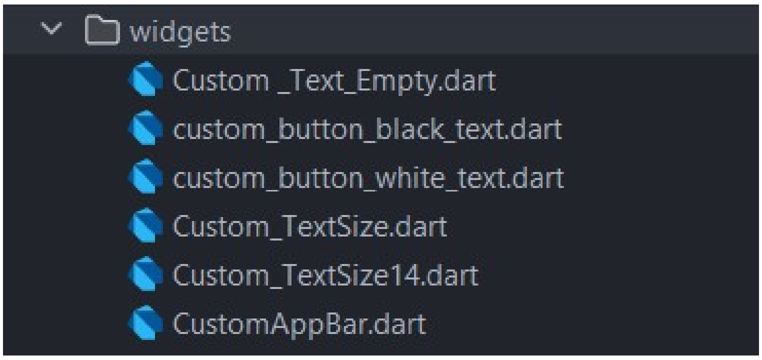


```
core
  constants
    colors.dart
  widgets
    Custom _Text_Empty.dart
    custom_button_black_text.dart
    custom_button_white_text.dart
    Custom_TextSize.dart
    Custom_TextSize14.dart
    CustomAppBar.dart
    text_form.dart
```

- The Core Having 1 File inside of Constants Folder

  1. Colors.dart : this file contains all the Colors with hexa code, which are used in the whole application.

Widgets Folder



```
widgets
    Custom _Text_Empty.dart
    custom_button_black_text.dart
    custom_button_white_text.dart
    Custom_TextSize.dart
    Custom_TextSize14.dart
    CustomAppBar.dart
```

- The Widgets Folder Having 7 Custom Widgets File.

1. Custom_Text_Empty.dart : this file contains a custom designed text with Align center parameter.

2. custom_button_black_text.dart : this file contains a custom designed Button with a black colour text.

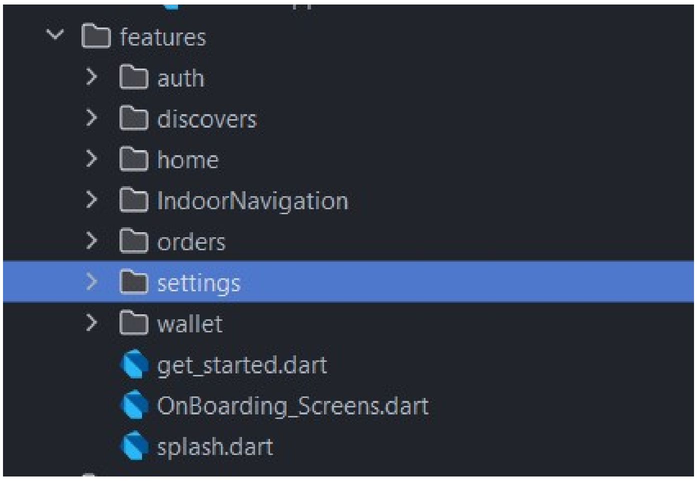
3. custom_button_white_text.dart : this file contains a custom designed Button with a white colour text.

4. Custom_TextSize.dart : this file contains a custom designed text.

5. Custom_TextSize14.dart : this file contains a custom designed text and with prefix 14 size.

6. CustomAppBar.dart : this file contains a custom designed AppBar.

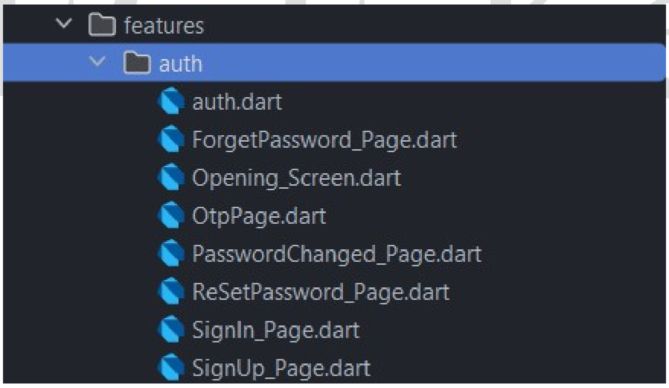
Features Folder



```
features
  auth
  discovers
  home
  IndoorNavigation
  orders
  settings
  wallet
  get_started.dart
  OnBoarding_Screens.dart
  splash.dart
```

- The Features Folder Having 7 Folders and 3 files.

1. Auth Folder



```
features
  auth
    auth.dart
    ForgetPassword_Page.dart
    Opening_Screen.dart
    OtpPage.dart
    PasswordChanged_Page.dart
    ReSetPassword_Page.dart
    SignIn_Page.dart
    SignUp_Page.dart
```

1. Auth.dart : this file contains the authentication functionalities code.

2. ForgetPassword_page.dart file : this file contains the Forget Password functionalities code.

3. Opening_Screen.dart file : this file contains the Opening Screen UI code.

4. OtpPage.dart file : this file contains the Otp Verification Functionalities code.

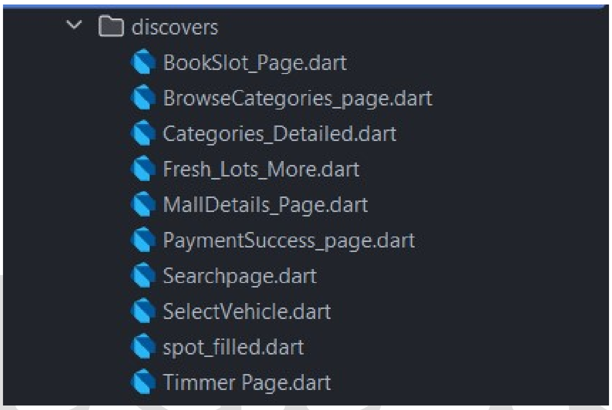
5. PasswordChanged_Page.dart file : this file contains the Password changed Successfully UI code.

6. ReSetPassword_Page.dart file : this file contains the Reset Password Functionalities code.

7. SignIn_Page.dart file : this file contains the SignIn Functionalities code.

8. SignUp_Page.dart file : this file contains the SignUp Functionalities code.

2. Discover Folder



```
discovers
    BookSlot_Page.dart
    BrowseCategories_page.dart
    Categories_Detailed.dart
    Fresh_Lots_More.dart
    MallDetails_Page.dart
    PaymentSuccess_page.dart
    Searchpage.dart
    SelectVehicle.dart
    spot_filled.dart
    Timmer Page.dart
```

1. BookSlot_Page.dart file : this file contains the Parking Lot Booking Functionalities code.

2. BrowseCategories_Page.dart file : this file contains the Browse Categories UI code.

3. Categories_Detailed.dart file : this file contains the All CategoriesUI code.

4. Fresh_Lots_More.dart file : this file contains the Fresh Lots in List(After Tapping on More Button) UI code.

5. MallDetails_Page.dart : this file contains the Mall Details (After Tapping on Mall Card) UI code.

6. PaymentSuccess_page.dart : this file contains the Payment Successfully done UI code.

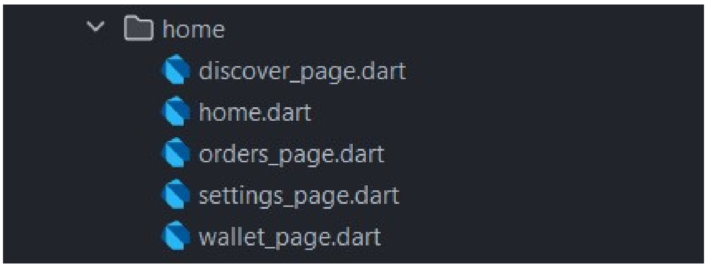
7. Searchpage.dart : this file contains the Search Functionalities and UI code.

8. SelectVehicle.dart file : this file contains the Select Vehicle Functionalities and UI code.

9. Spot_filled.dart file : this file contains the Check Out Spot Filled Functionalities and UI code.

10. Timmer Page.dart file : this file contains the Timer Functionalities and UI code.

3. Home Folder



```
home
    discover_page.dart
    home.dart
    orders_page.dart
    settings_page.dart
    wallet_page.dart
```

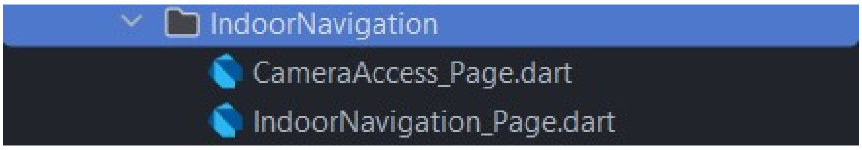
1. discover_page.dart file : this file contains the Discover Page Functionalities and UI code.

2. home.dart file : this file contains the Main Home Page of Application Functionalities and UI code.

3. orders_page.dart file : this file contains the Orders page Functionalities and UI code.
4. settings_page.dart : this file contains the Settings page Functionalities and UI code.

5. wallet_page.dart : this file contains the Wallet page Functionalities and UI code.

4. IndoorNavigation Folder

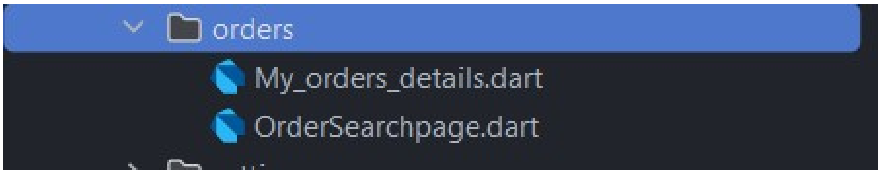


```
IndoorNavigation
    CameraAccess_Page.dart
    IndoorNavigation_Page.dart
```

1. CameraAccess_Page.dart file : this file contains the Camera Access Functionalities and UI code.

2. Indooravigation_Page.dart file : this file contains the Indoor Navigations Functionalities and UI code.

5. Orders folder

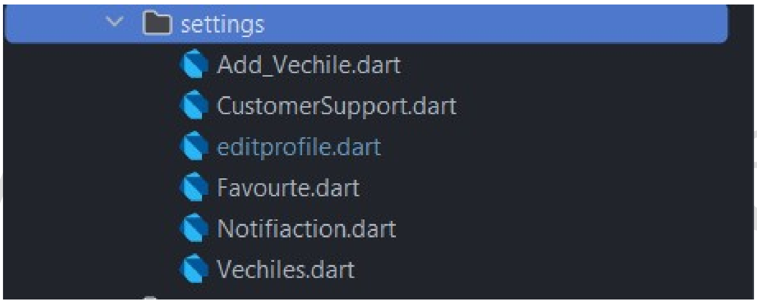


```
orders
    My_orders_details.dart
    OrderSearchpage.dart
```

1. My_orders_details.dart file : this file contains the Deatils of Orders Functionalities and UI code.

2. OrderSearchPage.dart file : this file contains the Orders Searching Functionalities and UI code.

6. Settings Folder



```
settings
    Add_Vechile.dart
    CustomerSupport.dart
    editprofile.dart
    Favourte.dart
    Notifiaction.dart
    Vechiles.dart
```

1. Add_Vehicle.dart file : this file contains the Add vehicle Functionalities and UI code.

2. CustomerSupport.dart file : this file contains the Customer Support UI code.

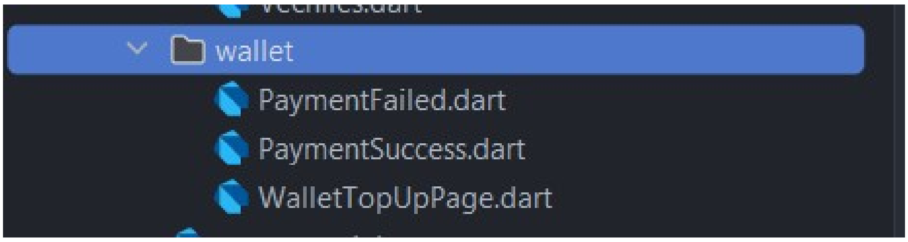
3. editprofile.dart : this file contains the Profile Editing Functionalities and UI code.

4. Favourate.dart : this file contains the Favourate Page Functionalities and UI code.

5. Notification.dart file : this file contains the Notification Functionalities and UI code.

6. Vehicles.dart : this file contains the Vehices Page Functionalities and UI code.

7. Wallet Folder



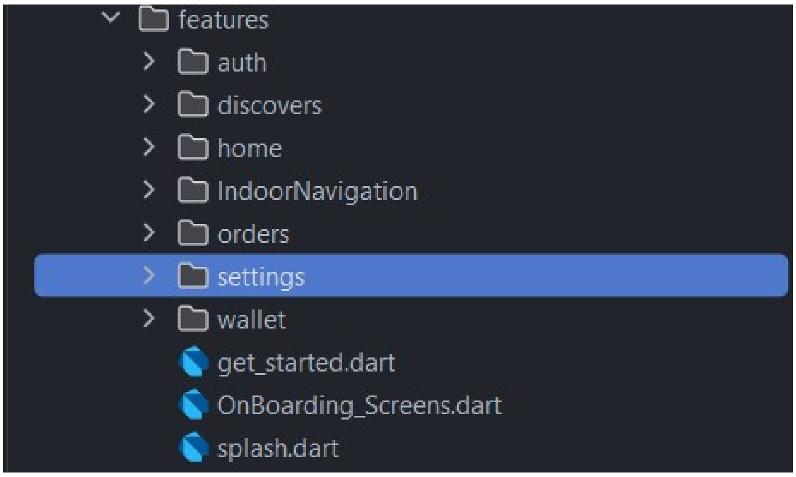
```
wallet
    PaymentFailed.dart
    PaymentSuccess.dart
    WalletTopUpPage.dart
```

1. PaymentFailed.dart : this file contains the Payment Failed UI code.

2. PaymentSuccess.dart : this file contains the Payment Success UI code.

3. WalletTopUpPage.dart : this file contains the Wallet Top Up Functionalities and UI code.

- Three Dart Files Are there in Features Folder

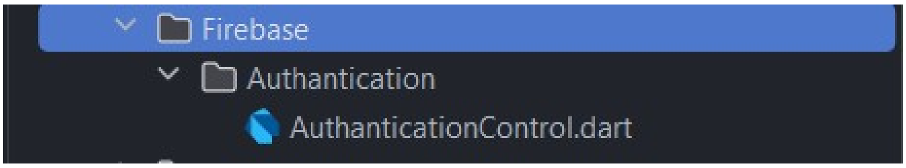


```
features
    auth
    discovers
    home
    IndoorNavigation
    orders
    settings
    wallet
    get_started.dart
    OnBoarding_Screens.dart
    splash.dart
```

1. get_started.dart : this file contains the Welcome Page UI code.
2. OnBoarding_Screens.dart : this file contains the Onboarding PagesUI code.

3. splash.dart : this file contains the Splash Screen UI code.

Firebase Folder

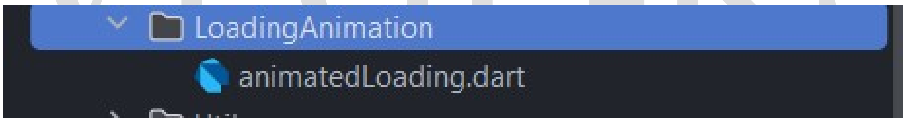


Firebase
Authantication
AuthanticationControl.dart

- The Features Folder Having 1 dart file inside of Authentication Folder.

1. AuthenticationControl.dart : this file contains the Authentication Control Functionalities code.

Loading Animation Folder

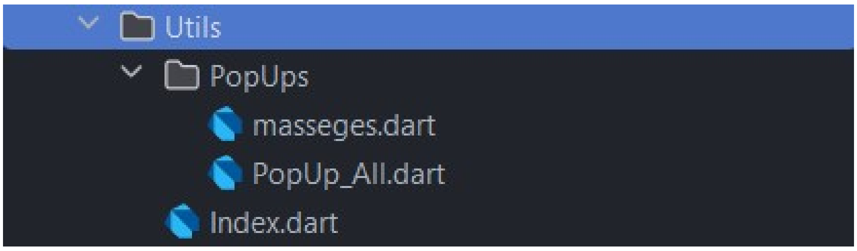


LoadingAnimation
animatedLoading.dart

- The Loading Animation Folder Having 1 dart file.

  1. animatedLoading.dart : this file contains the Loading Animation Functionalities and UI code.

Utils Folder



Utils
PopUps
masseges.dart
PopUp_All.dart
Index.dart

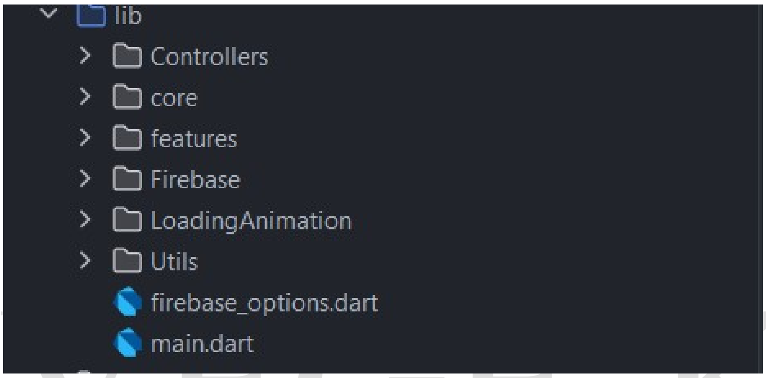
- The Utils Folder Having 1 Dart file and 1 Folder with 2 dart files.

1. massages.dart : this file contains the Messege in SnackBar UI code.

2. PopUp_All.dart : this file contains the All PopUps Functionalities and UI code.

3. Index.dart : this file contains the All index managing Functionalities code.

Lib Folder

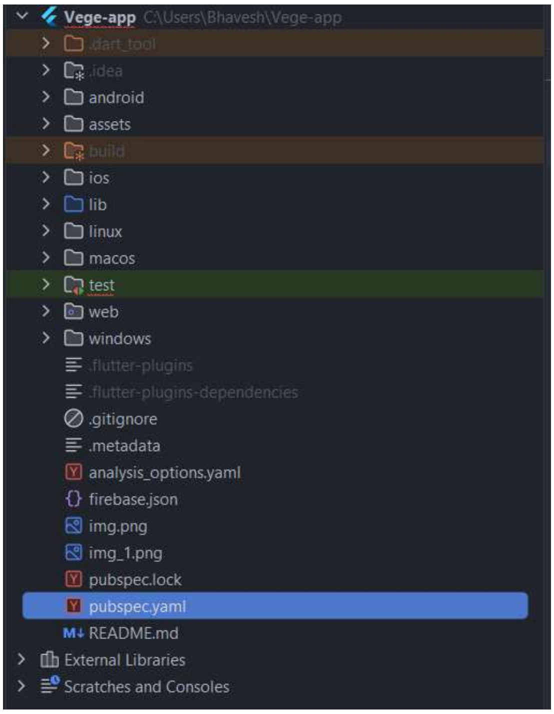


```
lib
  Controllers
  core
  features
  Firebase
  LoadingAnimation
  Utils
  firebase_options.dart
  main.dart
```

- The Lib Folder having 2 dart files.
    1. firebase_option.dart : this file contains the Firebase Information Functionalities code.

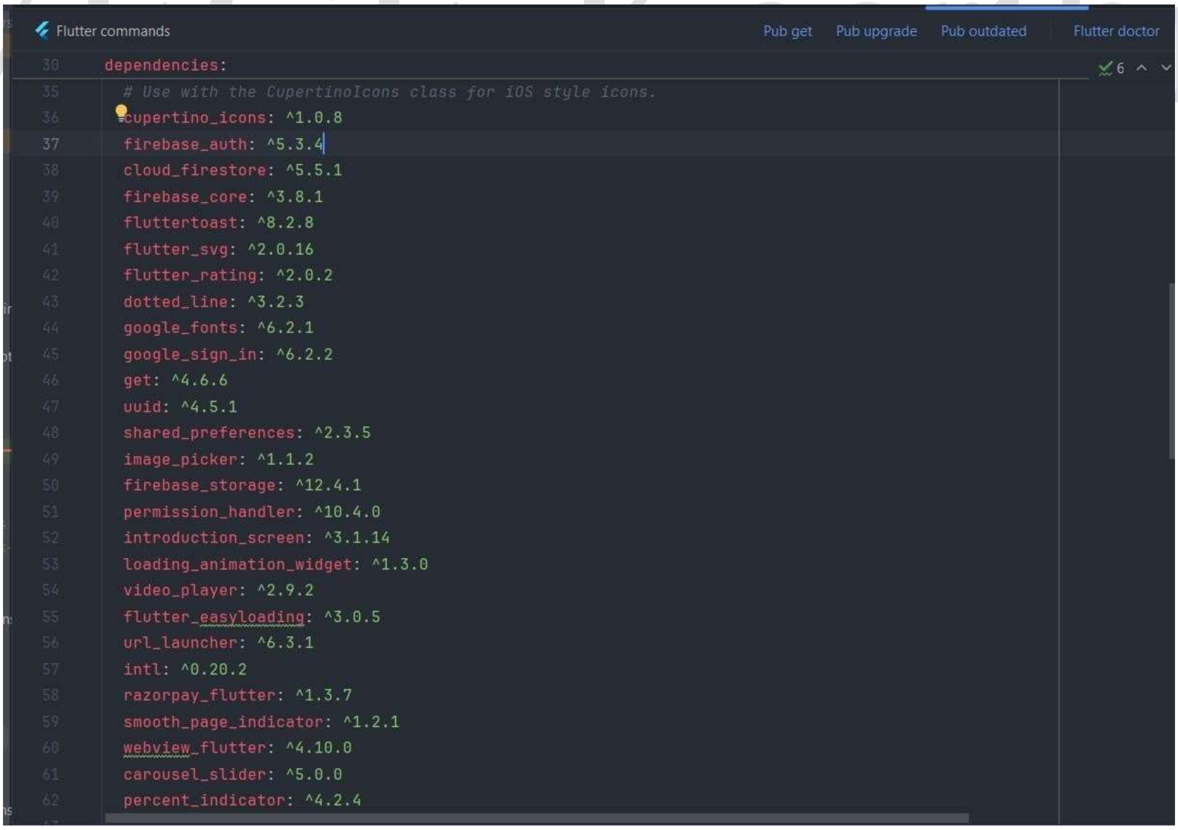
    2. main.dart : this file contains the Startup of Application and firebase Functionalities code.

pubspec.yaml file



```
Vege-app C:\Users\Bhavesh\Vege-app
  .dart_tool
  .idea
  android
  assets
  build
  ios
  lib
  linux
  macos
  test
  web
  windows
  .flutter-plugins
  .flutter-plugins-dependencies
  .gitignore
  .metadata
  analysis_options.yaml
  firebase.json
  img.png
  img_1.png
  pubspec.lock
  pubspec.yaml
  README.md
External Libraries
Scratches and Consoles
```

- The pubspec.yaml file contains the all imported packages.



```yaml
dependencies:
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  firebase_auth: ^5.3.4
  cloud_firestore: ^5.5.1
  firebase_core: ^3.8.1
  fluttertoast: ^8.2.8
  flutter_svg: ^2.0.16
  flutter_rating: ^2.0.2
  dotted_line: ^3.2.3
  google_fonts: ^6.2.1
  google_sign_in: ^6.2.2
  get: ^4.6.6
  uuid: ^4.5.1
  shared_preferences: ^2.3.5
  image_picker: ^1.1.2
  firebase_storage: ^12.4.1
  permission_handler: ^10.4.0
  introduction_screen: ^3.1.14
  loading_animation_widget: ^1.3.0
  video_player: ^2.9.2
  flutter_easyloading: ^3.0.5
  url_launcher: ^6.3.1
  intl: ^0.20.2
  razorpay_flutter: ^1.3.7
  smooth_page_indicator: ^1.2.1
  webview_flutter: ^4.10.0
  carousel_slider: ^5.0.0
  percent_indicator: ^4.2.4
```

pubspec.yaml File

Contains all imported external packages and dependencies:

- **cupertino_icons**: iOS-style icons.
- **firebase_auth**: Firebase authentication.
- **cloud_firestore**: Firebase Firestore database operations.
- **firebase_core**: Firebase initialization.
- **fluttertoast**: Toast notifications.
- **flutter_svg**: Display SVG images.
- **flutter_rating**: Rating widgets.
- **dotted_line**: Dotted or dashed line separators.
- **google_fonts**: Google Fonts integration.
- **google_sign_in**: Google Sign-In integration.
- **get**: State management, routing, dependency injection.
- **uuid**: Unique ID generation.
- **shared_preferences**: Local data storage.
- **image_picker**: Pick images from gallery or camera.
- **firebase_storage**: Store files in Firebase Storage.
- **permission_handler**: Request/manage permissions.
- **introduction_screen**: Onboarding screen setup.
- **loading_animation_widget**: Custom loading animations.

- **flutter_easyloading**: Stylish loading indicators.
- **url_launcher**: Launch URLs (web, phone, email, etc.).
- **intl**: Internationalization and localization.
- **razorpay_flutter**: Razorpay payment gateway.
- **smooth_page_indicator**: Page indicators for swiping screens.
- **webview_flutter**: Web page embedding in app.
- **carousel_slider**: Image sliders/carousels.
- **percent_indicator**: Circular and linear percent indicators.

Step 3.1: Flutter App Development

- **Modules**:
  - Login/Register (Firebase Auth)
  - Slot Reservation Screen
  - Indoor Navigation using coordinates
  - Wallet + Payment System
  - Admin Dashboard for analytics

Step 3.2: Firebase Backend

- Firestore: Users, Reservations, Transactions, Analytics
- Realtime DB: Live Slot Updates, Entry/Exit Logs
- Firebase Functions: Auto-cancel if not entered in 15 mins

Step 3.3: License Plate Recognition (LPR)

- Use ESP32-CAM to send image to OpenALPR
- Extract license number, validate via Firebase

Phase 4: Testing & Validation

Step 4.1: Unit Testing

- Flutter app screens, Firebase integration
- Servo response to plate number
- NodeMCU sensor accuracy

Step 4.2: Integration Testing

- End-to-end test from vehicle entry to slot assignment
- Wallet deduction based on entry and exit times

Step 4.3: Edge Cases

- Invalid plates
- Time extension logic (15 min → 20 min x6/month)
- Double bookings

Phase 5: Deployment & Monitoring

Step 5.1: Final Integration

- Deploy hardware at entry point and slots
- Ensure real-time sync with Firebase

Step 5.2: Admin Setup

- Admin dashboard for:
  - Monitoring active users
  - Viewing transactions
  - Analytics: Peak hours, total revenue

Step 5.3: User Training

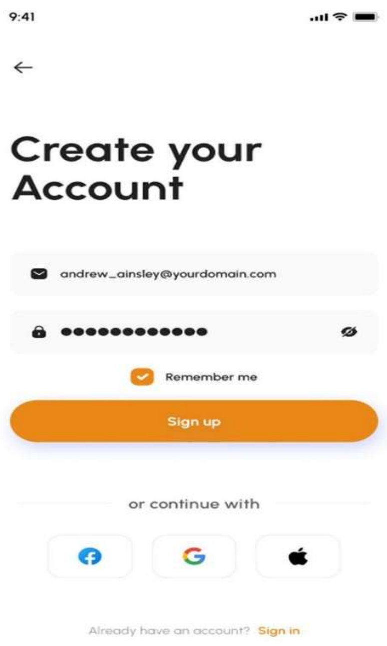
- User guide to reserve, pay, and use navigation

Phase 6: Maintenance and Feedback

Step 6.1: Monitor Logs

- Firebase Logs
- Sensor failure or connectivity issues

3. Screenshots of Execution



9:41

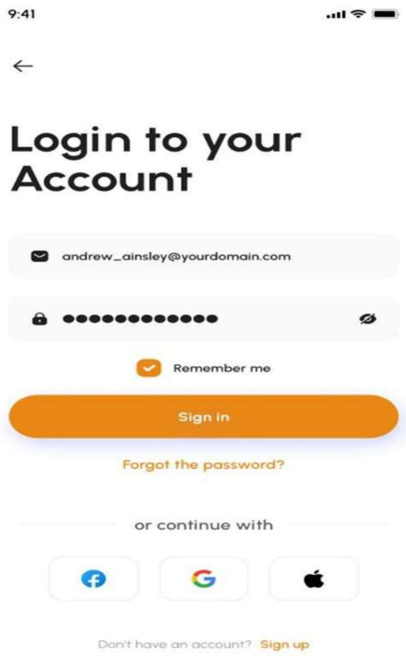
Create your Account

andrew_ainsley@yourdomain.com

●●●●●●●●●●●●●

☑ Remember me

Already have an account? Sign in



9:41

Login to your Account

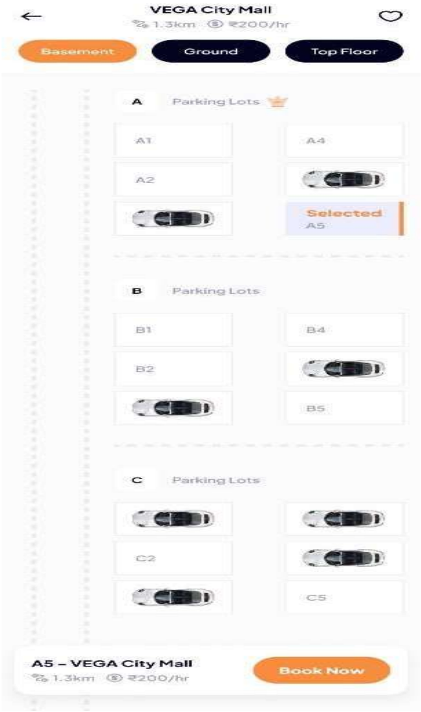
andrew_ainsley@yourdomain.com

●●●●●●●●●●●●●

☑ Remember me

Don't have an account? Sign up

Fig 1. and Fig 2. Illustrates the Sign in and sign-up screens in the app



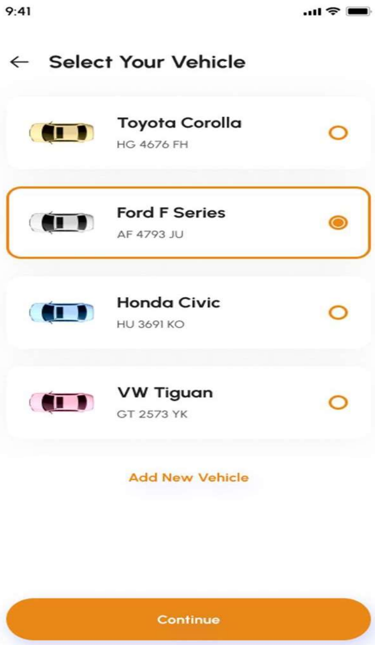
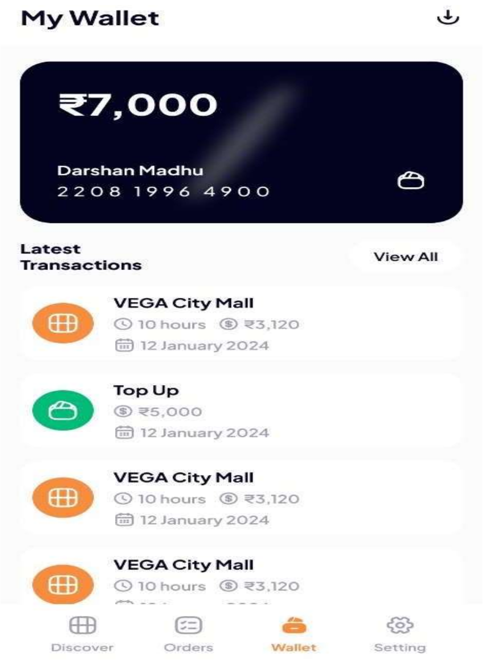


Fig 3. User Slot Selection

Fig 4. Select Your Vehicle Option



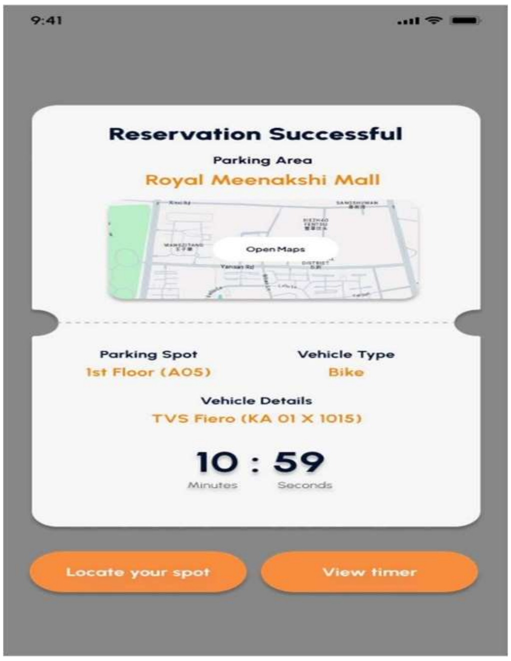


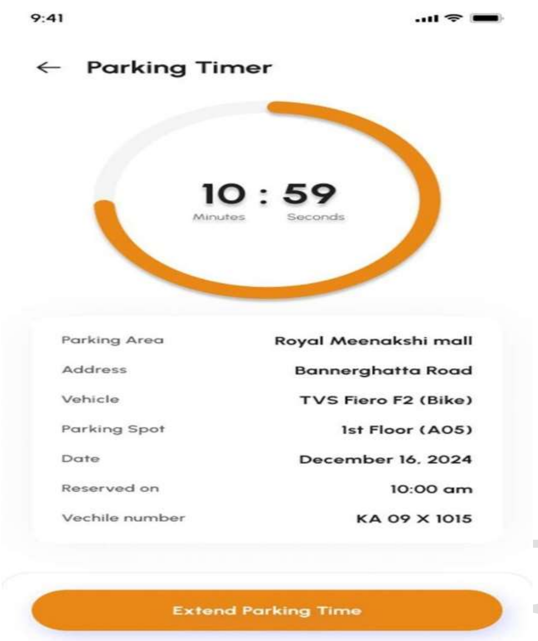
Fig 5. Illustrates the wallet for

for automatic payment

Fig 6. Illustrates that the reservation

is successful and the timer starts

for 15min



9:41

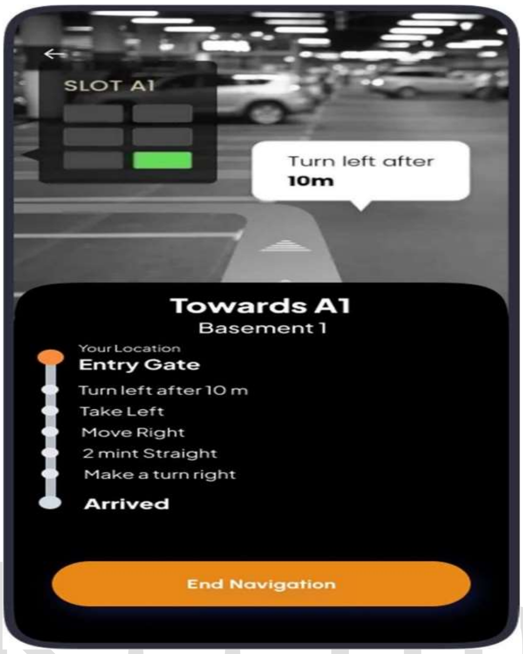
← Parking Timer

10 : 59
Minutes Seconds

Parking Area	Royal Meenakshi mall
Address	Bannerghatta Road
Vehicle	TVS Fiero F2 (Bike)
Parking Spot	1st Floor (A05)
Date	December 16, 2024
Reserved on	10:00 am
Vechile number	KA 09 X 1015

Fig 7. Parking Timer for 15 min for the

vehicle to enter

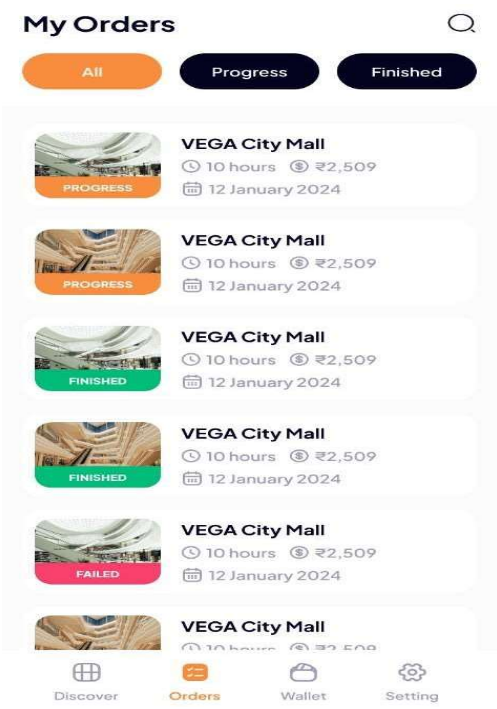


Towards A1
Basement 1

Your Location
Entry Gate
Turn left after 10 m
Take Left
Move Right
2 mint Straight
Make a turn right
Arrived

Fig 8. Indoor Navigation after the

vehicle enters the parking basement



My Orders

VEGA City Mall
10 hours ₹2,509
12 January 2024
PROGRESS

VEGA City Mall
10 hours ₹2,509
12 January 2024
PROGRESS

VEGA City Mall
10 hours ₹2,509
12 January 2024
FINISHED

VEGA City Mall
10 hours ₹2,509
12 January 2024
FINISHED

VEGA City Mall
10 hours ₹2,509
12 January 2024
FAILED

VEGA City Mall

Fig 8. Shows my orders in the app

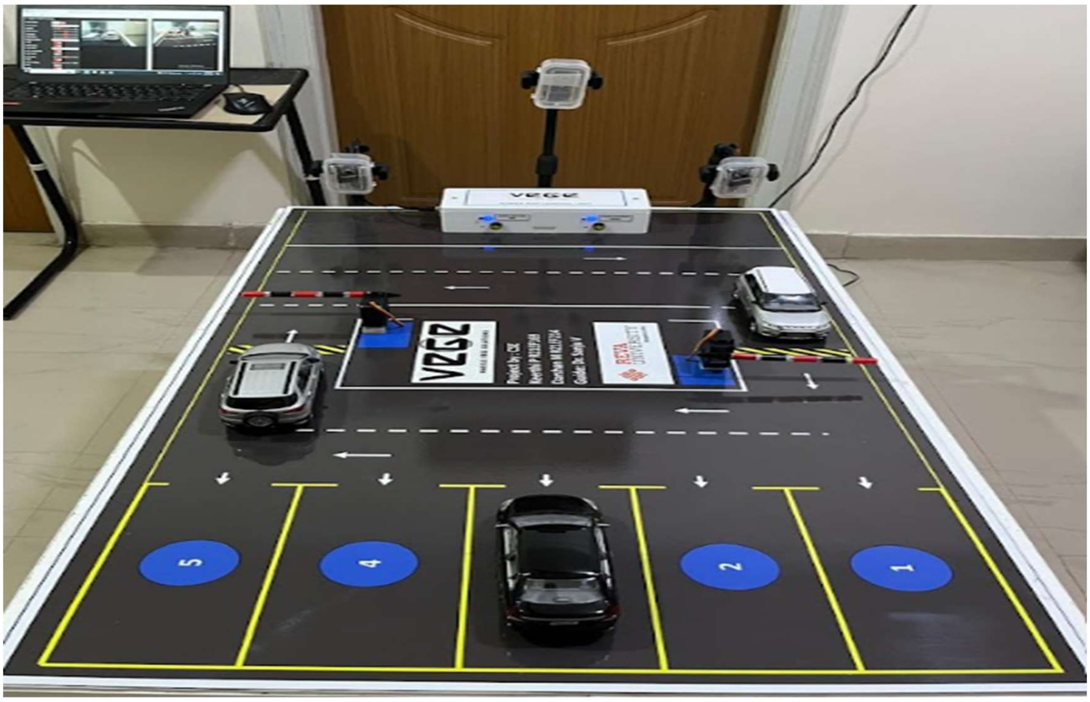


Fig 9. Illustrates the Prototype model with connections and cameras

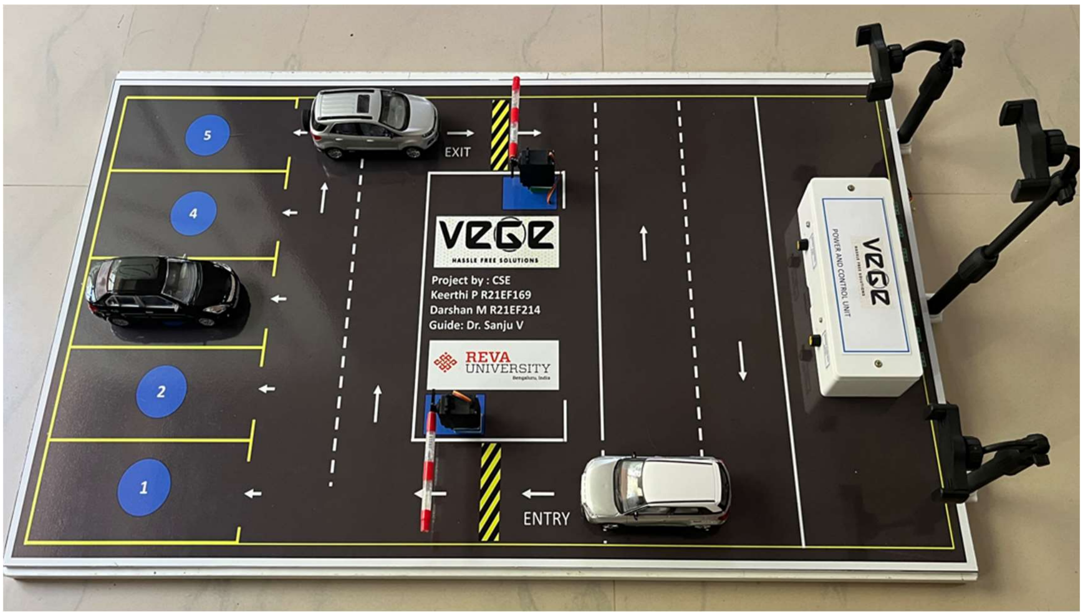


Fig 10. Illustrates the prototype model