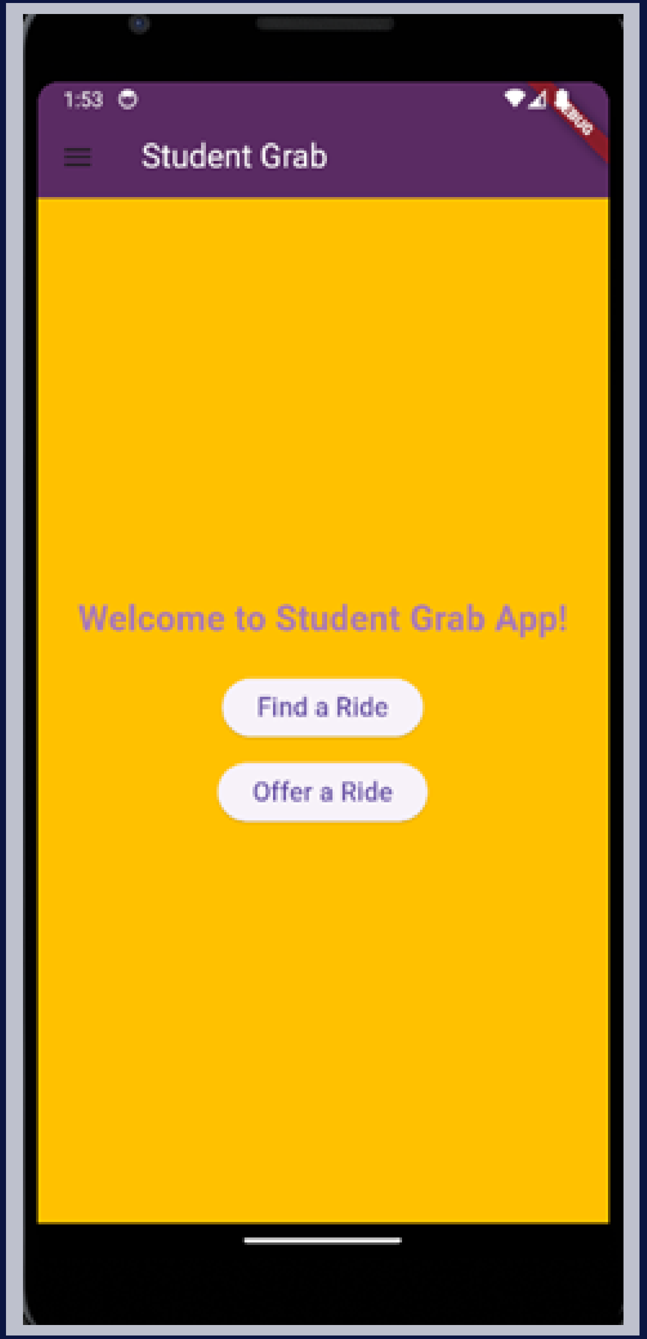
STUDENT GRAB APP

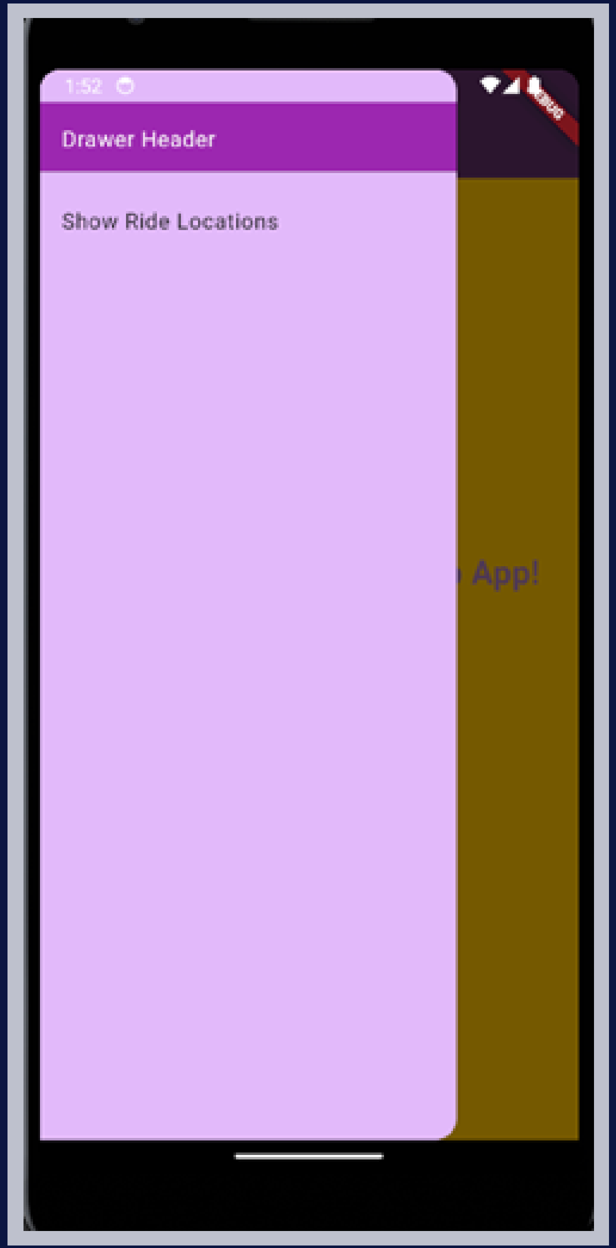
The purpose of the Student Grab Application is to improve student mobility by offering a safe and practical way to get around. The application seeks to make the commuting experience for students safer in response to safety concerns expressed by students, such as assault and kidnapping events. The platform is open to university students possessing a valid driver's license, encouraging them to register as drivers or passengers. By leveraging the app, students can enjoy a reliable and secure transportation service tailored to their specific needs.

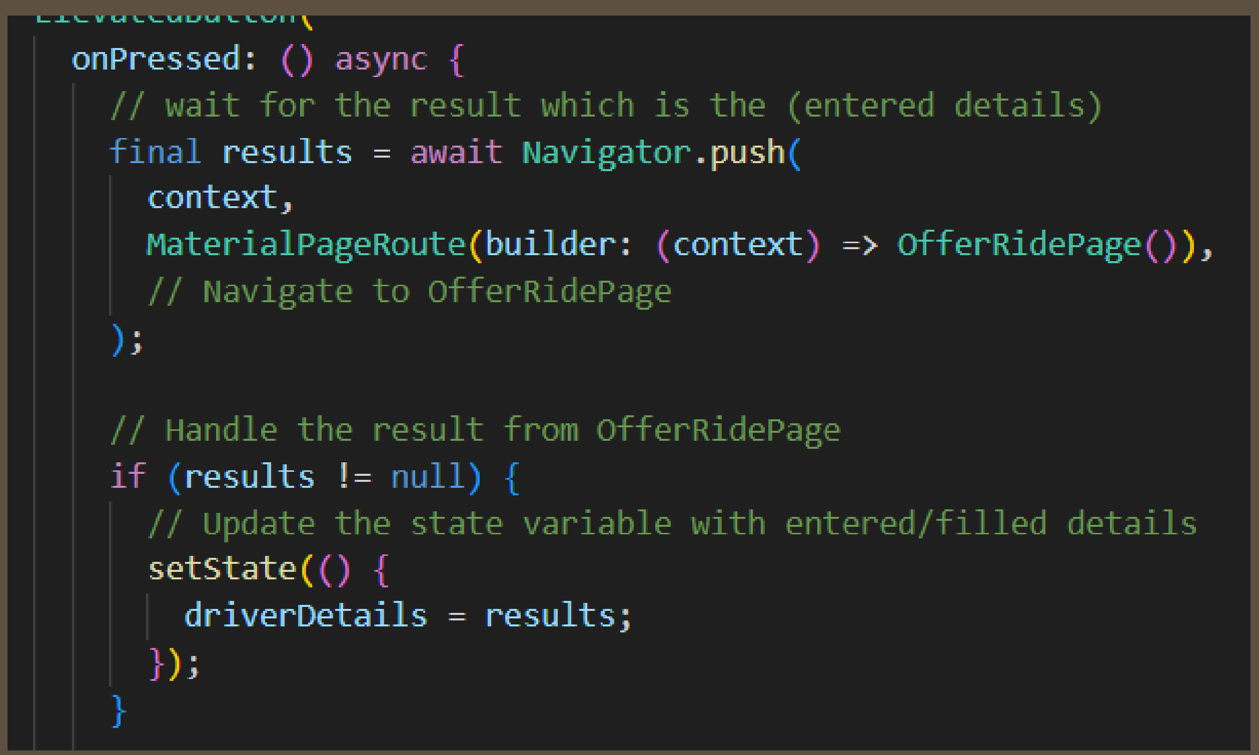
prepared: Nur Elya Farhana binti
Zainordin
S63723

HOMEPAGE

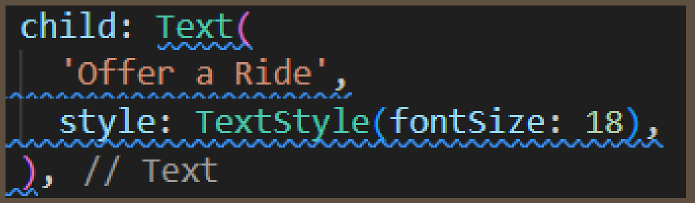
It is the main page. User may explore through this page by clicking the text or button to access too other pages.



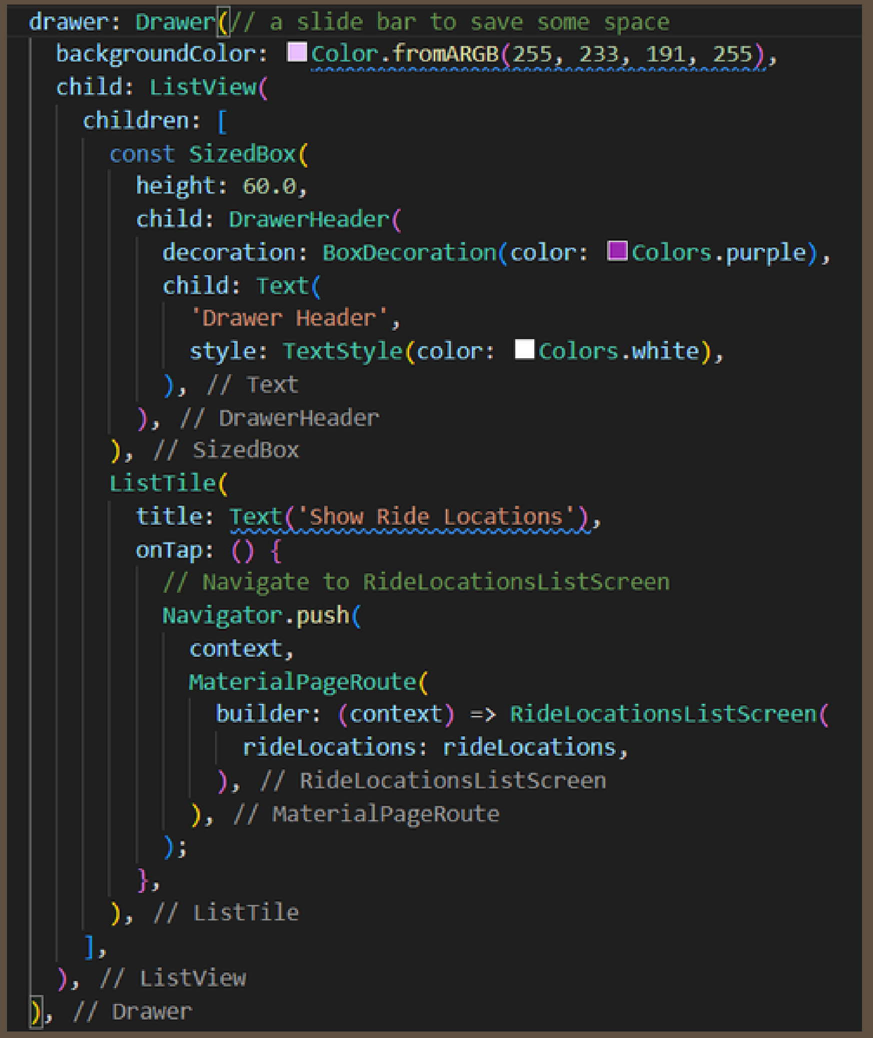




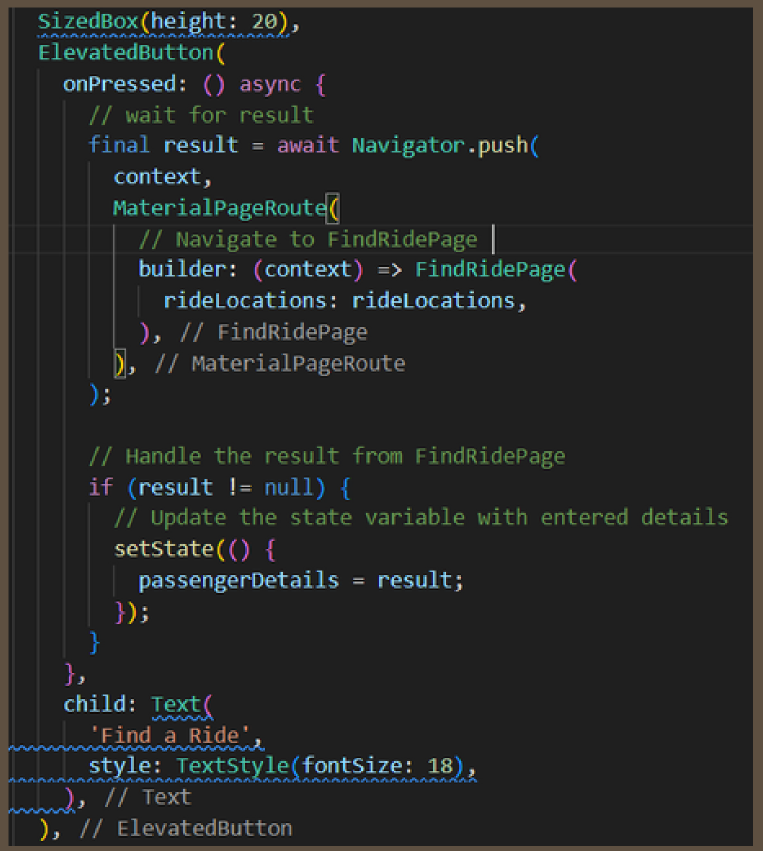
```
ElevatedButton(
  onPressed: () async {
    // wait for the result which is the (entered details)
    final results = await Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => OfferRidePage()),
      // Navigate to OfferRidePage
    );

    // Handle the result from OfferRidePage
    if (results != null) {
      // Update the state variable with entered/filled details
      setState(() {
        driverDetails = results;
      });
    }
```



```
child: Text(
  'Offer a Ride',
  style: TextStyle(fontSize: 18),
), // Text
```



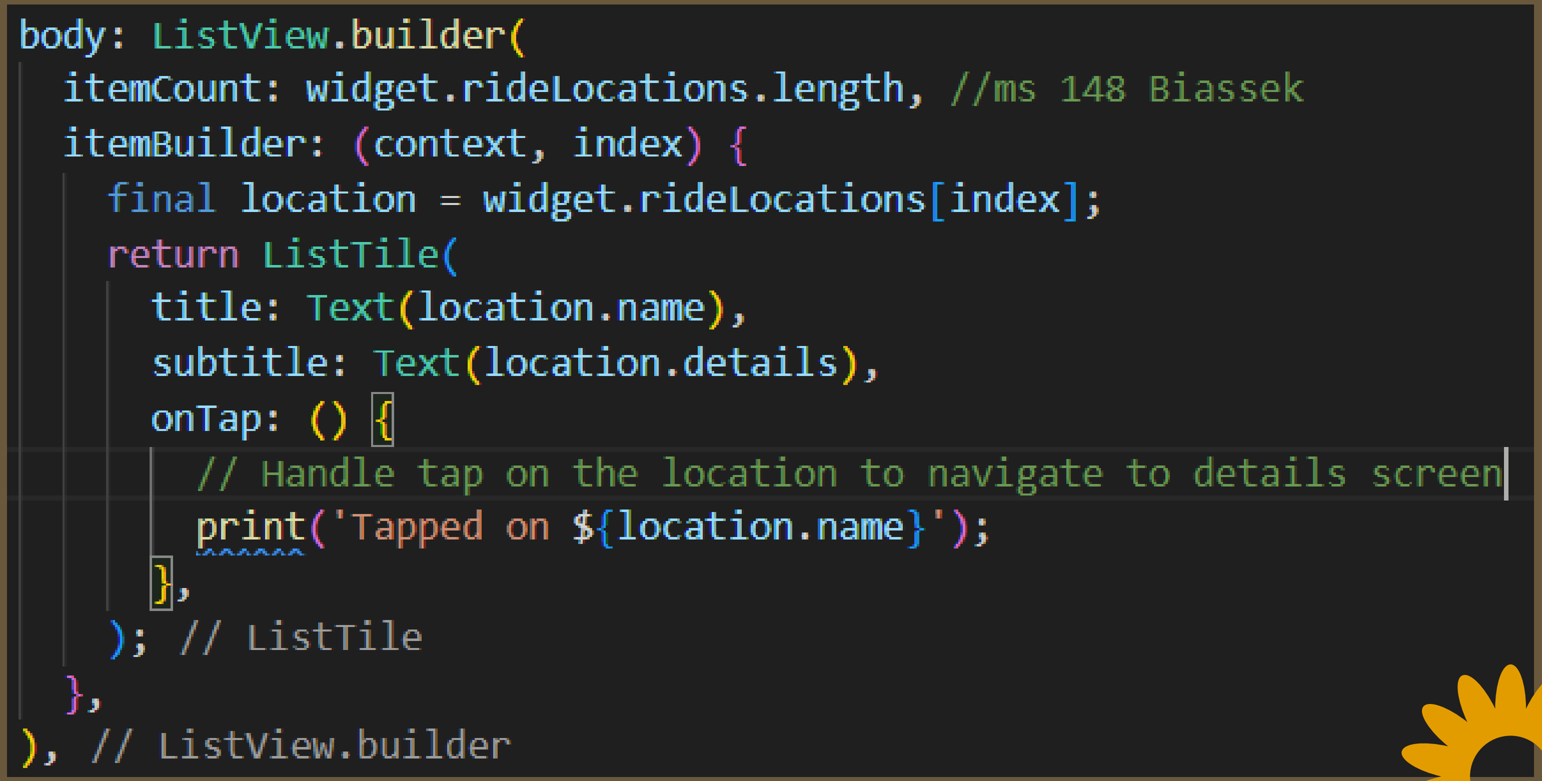
```
drawer: Drawer(// a slide bar to save some space
  backgroundColor: Color.fromARGB(255, 233, 191, 255),
  child: ListView(
    children: [
      const SizedBox(
        height: 60.0,
        child: DrawerHeader(
          decoration: BoxDecoration(color: Colors.purple),
          child: Text(
            'Drawer Header',
            style: TextStyle(color: Colors.white),
          ), // Text
        ), // DrawerHeader
      ), // SizedBox
      ListTile(
        title: Text('Show Ride Locations'),
        onTap: () {
          // Navigate to RideLocationsListScreen
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => RideLocationsListScreen(
                rideLocations: rideLocations,
              ), // RideLocationsListScreen
            ), // MaterialPageRoute
          );
        },
      ), // ListTile
    ],
  ), // ListView
), // Drawer
```



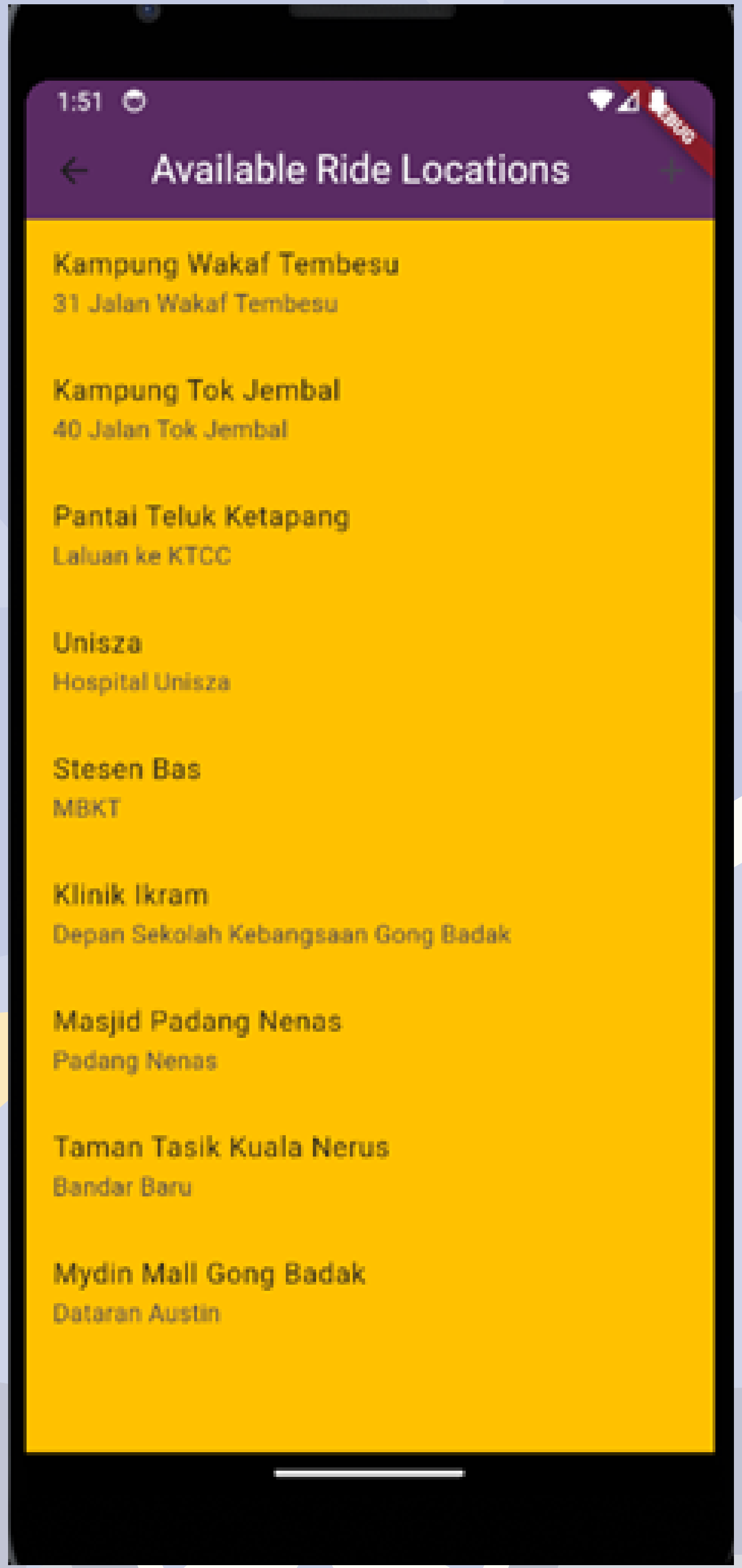
```
SizedBox(height: 20),
ElevatedButton(
  onPressed: () async {
    // wait for result
    final result = await Navigator.push(
      context,
      MaterialPageRoute(
        // Navigate to FindRidePage
        builder: (context) => FindRidePage(
          rideLocations: rideLocations,
        ), // FindRidePage
      ), // MaterialPageRoute
    );

    // Handle the result from FindRidePage
    if (result != null) {
      // Update the state variable with entered details
      setState(() {
        passengerDetails = result;
      });
    }
  },
  child: Text(
    'Find a Ride',
    style: TextStyle(fontSize: 18),
  ), // Text
), // ElevatedButton
```

AVAILABLE RIDE LOCATIONS PAGE

This page lists all the available ride locations.



```
body: ListView.builder(
  itemCount: widget.rideLocations.length, //ms 148 Biassek
  itemBuilder: (context, index) {
    final location = widget.rideLocations[index];
    return ListTile(
      title: Text(location.name),
      subtitle: Text(location.details),
      onTap: () {
        // Handle tap on the location to navigate to details screen
        print('Tapped on ${location.name}');
      },
    ); // ListTile
  },
), // ListView.builder
```



Available Ride Locations

Kampung Wakaf Tembesu
31 Jalan Wakaf Tembesu

Kampung Tok Jembal
40 Jalan Tok Jembal

Pantai Teluk Ketapang
Laluan ke KTCC

Unisza
Hospital Unisza

Stesen Bas
MBKT

Klinik Ikram
Depan Sekolah Kebangsaan Gong Badak

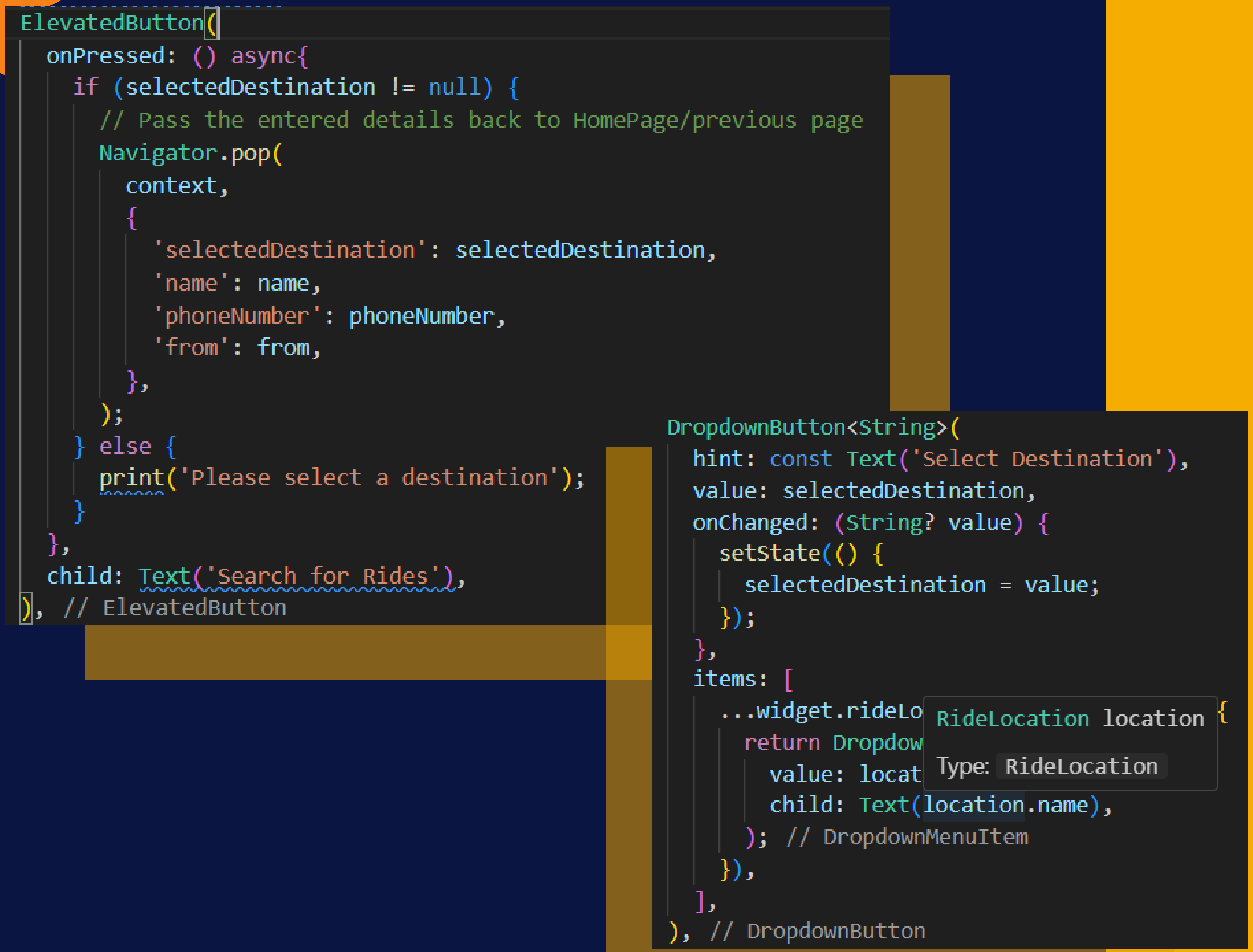
Masjid Padang Nenas
Padang Nenas

Taman Tasik Kuala Nerus
Bandar Baru

Mydin Mall Gong Badak
Dataran Austin

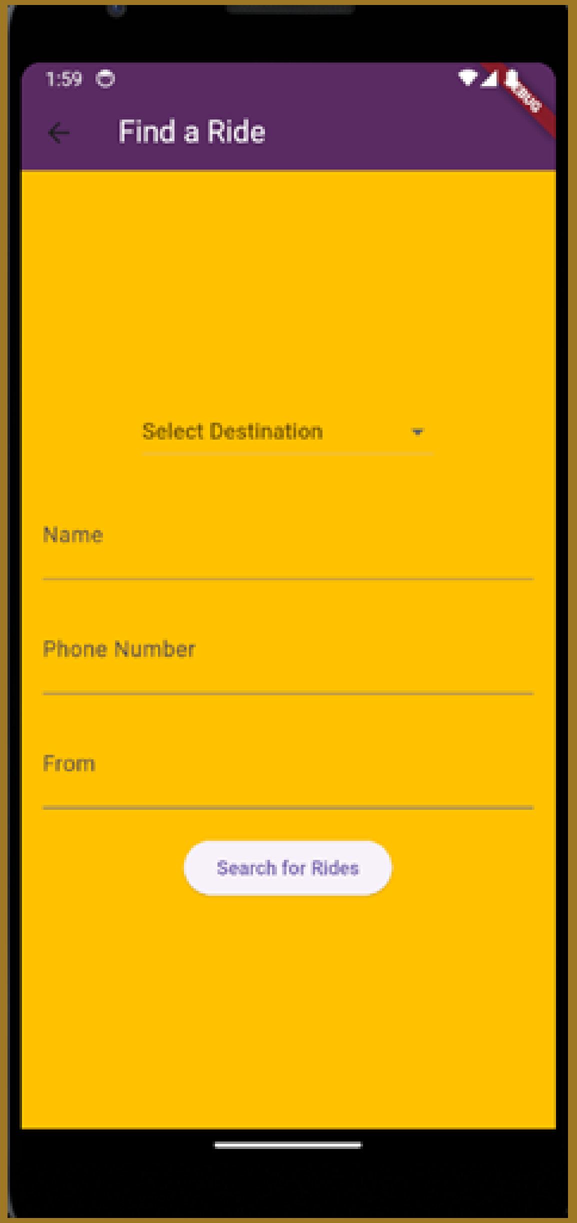
FIND A RIDE PAGE

This is the page that user is required to fill in the details to find a ride



```
ElevatedButton(
  onPressed: () async{
    if (selectedDestination != null) {
      // Pass the entered details back to HomePage/previous page
      Navigator.pop(
        context,
        {
          'selectedDestination': selectedDestination,
          'name': name,
          'phoneNumber': phoneNumber,
          'from': from,
        },
      );
    } else {
      print('Please select a destination');
    }
  },
  child: Text('Search for Rides'),
), // ElevatedButton
```

```
DropdownButton<String>(
  hint: const Text('Select Destination'),
  value: selectedDestination,
  onChanged: (String? value) {
    setState(() {
      selectedDestination = value;
    });
  },
  items: [
    ...widget.rideLo RideLocation location {
      return Dropdow
        value: locat Type: RideLocation
        child: Text(location.name),
      ); // DropdownMenuItem
    }),
  ],
), // DropdownButton
```



1:59

Find a Ride

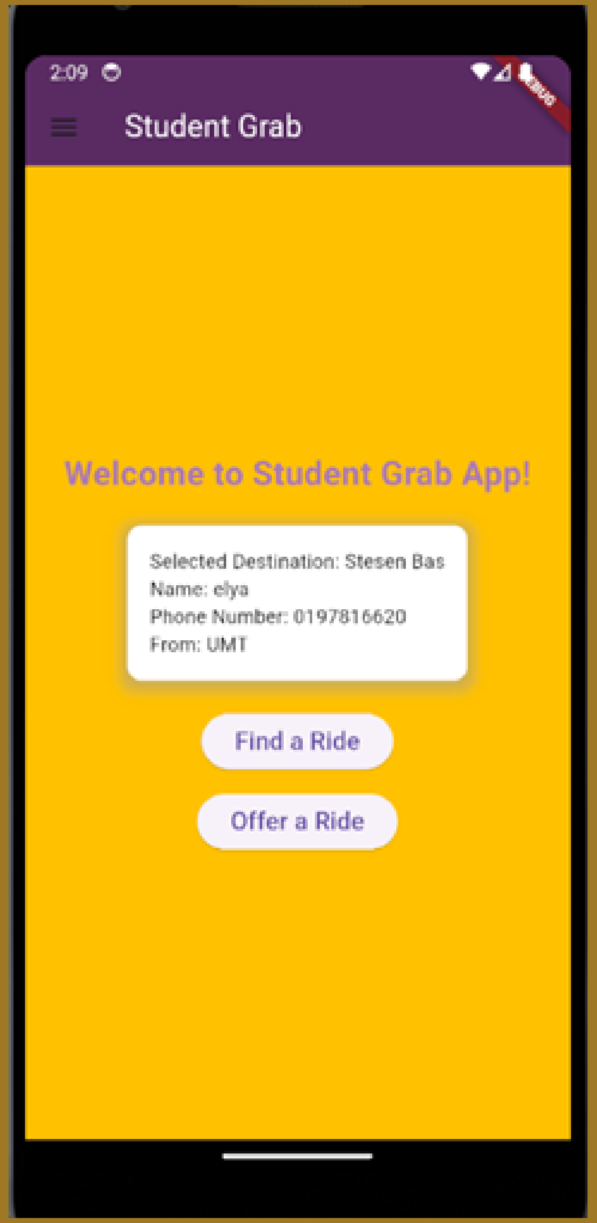
Select Destination

Name

Phone Number

From

Search for Rides



2:09

Student Grab

Welcome to Student Grab App!

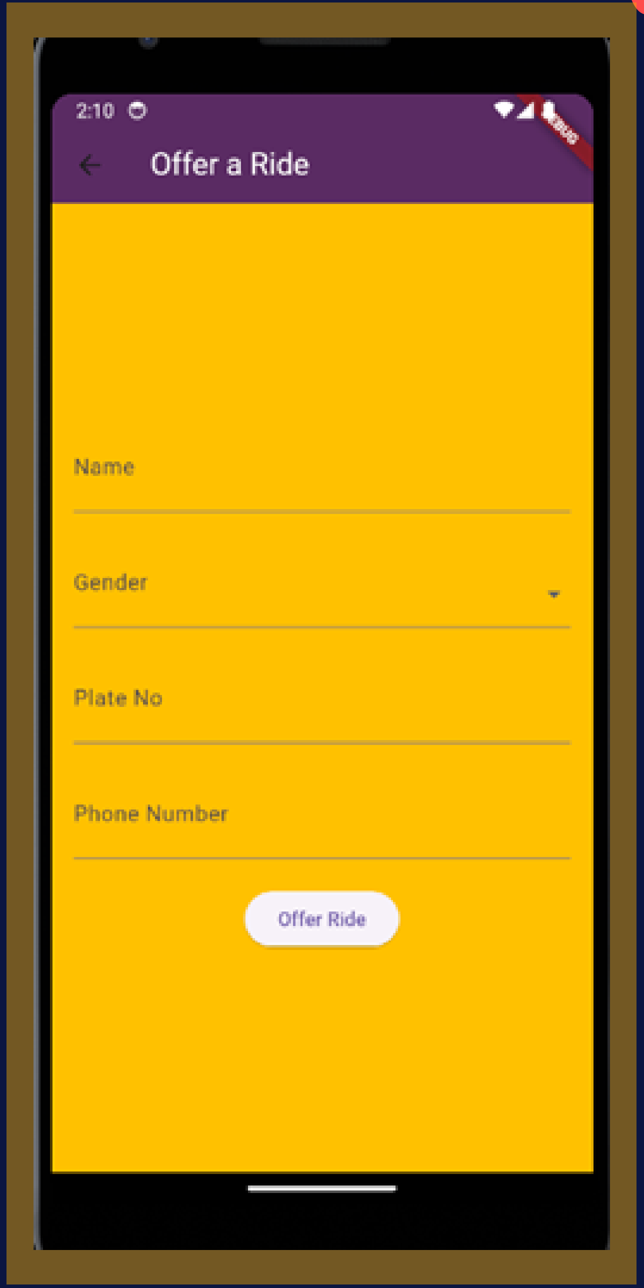
Selected Destination: Stesen Bas
Name: elya
Phone Number: 0197816620
From: UMT

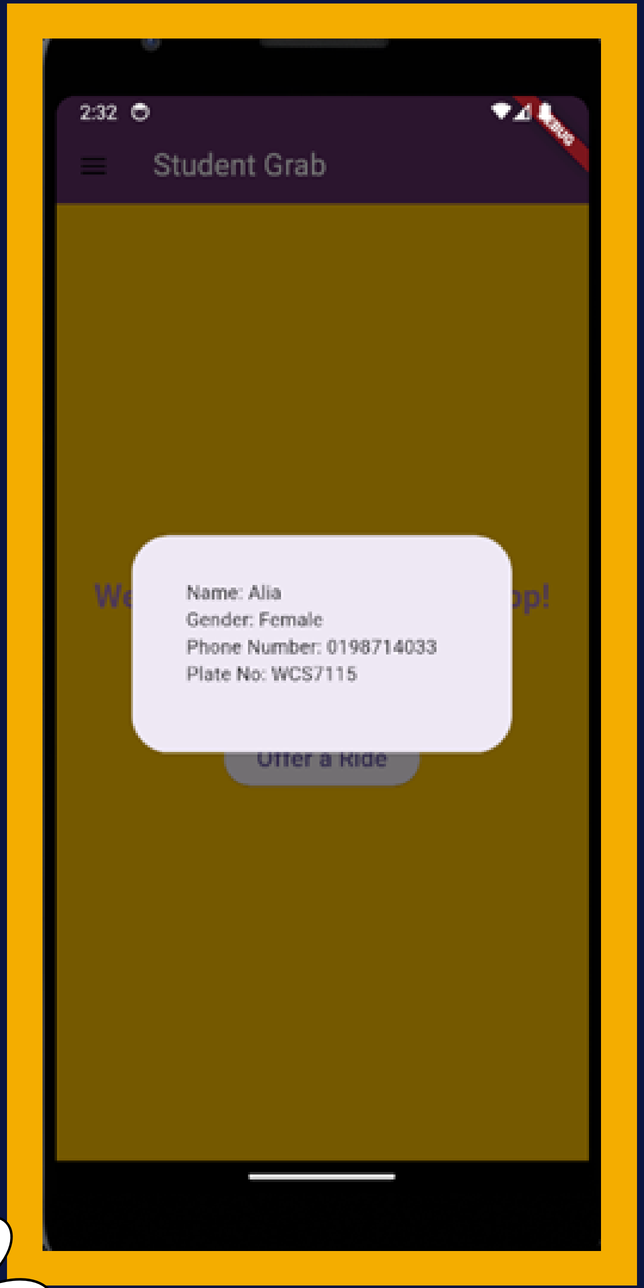
Find a Ride

Offer a Ride

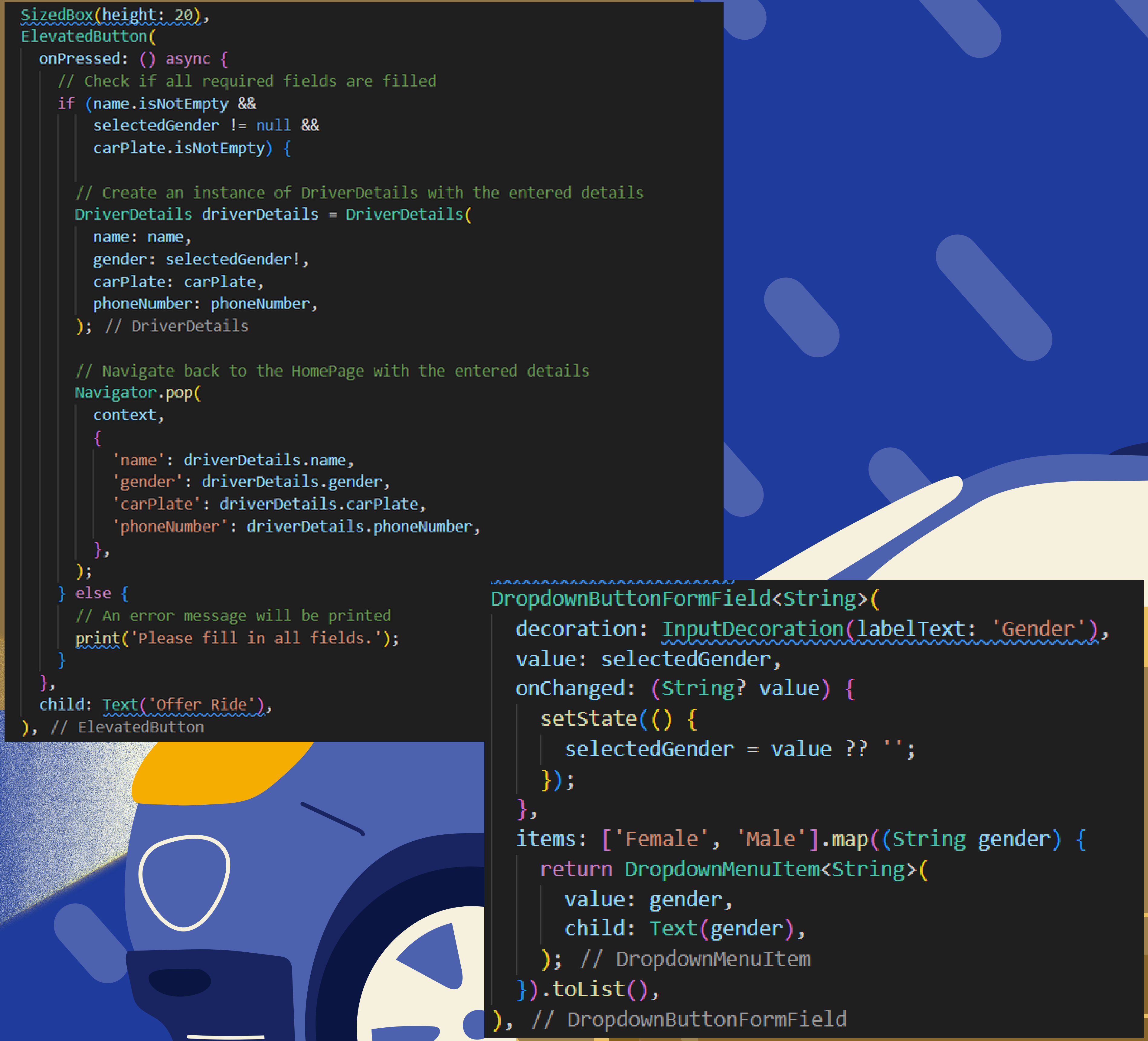
OFFER A RIDE PAGE

This is where the user has to fill in the details to offer a ride.





Name: Alia
Gender: Female
Phone Number: 0198714033
Plate No: WCS7115



```
SizedBox(height: 20),
ElevatedButton(
  onPressed: () async {
    // Check if all required fields are filled
    if (name.isNotEmpty &&
        selectedGender != null &&
        carPlate.isNotEmpty) {

      // Create an instance of DriverDetails with the entered details
      DriverDetails driverDetails = DriverDetails(
        name: name,
        gender: selectedGender!,
        carPlate: carPlate,
        phoneNumber: phoneNumber,
      ); // DriverDetails

      // Navigate back to the HomePage with the entered details
      Navigator.pop(
        context,
        {
          'name': driverDetails.name,
          'gender': driverDetails.gender,
          'carPlate': driverDetails.carPlate,
          'phoneNumber': driverDetails.phoneNumber,
        },
      );
    } else {
      // An error message will be printed
      print('Please fill in all fields.');
    }
  },
  child: Text('Offer Ride'),
), // ElevatedButton
```

```
DropdownButtonFormField<String>(
  decoration: InputDecoration(labelText: 'Gender'),
  value: selectedGender,
  onChanged: (String? value) {
    setState(() {
      selectedGender = value ?? '';
    });
  },
  items: ['Female', 'Male'].map((String gender) {
    return DropdownMenuItem<String>(
      value: gender,
      child: Text(gender),
    ); // DropdownMenuItem
  }).toList(),
), // DropdownButtonFormField
```