```
    },
    child: YourWidget(),
  )
```

CODE:

```
import 'dart:convert';

import 'dart:io';

import 'package:flutter/material.dart';

void main() {

  runApp(MyApp());

}

class Student {

  String name;

  String address;

  String mobileNumber;

  String email;

  String prn;

  String academicYear;

  Student({

    required this.name,

    required this.address,

    required this.mobileNumber,
```

```
    required this.email,
    required this.prn,
    required this.academicYear,
  });
 Map<String, dynamic> toJson() {
    return {
      'name': name,
      'address': address,
      'mobileNumber': mobileNumber,
      'email': email,
      'prn': prn,
      'academicYear': academicYear,
    };
  }
factory Student.fromJson(Map<String, dynamic> json) {
    return Student(
      name: json['name'],
      address: json['address'],
      mobileNumber: json['mobileNumber'],
      email: json['email'],
```

```
      prn: json['prn'],
      academicYear: json['academicYear'],
    );
  }
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Student Management App',
      home: MyHomePage(),
      theme: ThemeData.dark()
);
  }
}
class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}
```

```
class _MyHomePageState extends State<MyHomePage> {
  List<Student> students = [];
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  late TextEditingController _nameController;
  late TextEditingController _addressController;
  late TextEditingController _mobileController;
  late TextEditingController _emailController;
  late TextEditingController _prnController;
  String _selectedAcademicYear = 'First year';
  @override
  void initState() {
    super.initState();
    _nameController = TextEditingController();
    _addressController = TextEditingController();
    _mobileController = TextEditingController();
    _emailController = TextEditingController();
    _prnController = TextEditingController();
    _loadData();
  }
  void _loadData() async {
```

```
  try {
    File file = File(await _getFilePath());
    if (file.existsSync()) {
      List<dynamic> jsonData = json.decode(file.readAsStringSync());
      setState(() {
        students = jsonData.map((data) => Student.fromJson(data)).toList();
      });
    }
  } catch (e) {
    print("Error loading data: $e");
  }
}
void _saveData() async {
  File file = File(await _getFilePath());
  file.writeAsStringSync(json.encode(students));
}
Future<String> _getFilePath() async {
  final directory = await getApplicationDocumentsDirectory();
  return directory.path + '/students.json';
}
```

```
void _addStudent() {
  if (_formKey.currentState!.validate()) {
    setState(() {
      students.add(
        Student(
          name: _nameController.text,
          address: _addressController.text,
          mobileNumber: _mobileController.text,
          email: _emailController.text,
          prn: _prnController.text,
          academicYear: _selectedAcademicYear,
        ),
      );
      _saveData();
      _clearForm();
    });
  }
}
void _modifyStudent(int index) {
  if (_formKey.currentState!.validate()) {
```

```
      setState(() {
        students[index] = Student(
          name: _nameController.text,
          address: _addressController.text,
          mobileNumber: _mobileController.text,
          email: _emailController.text,
          prn: _prnController.text,
          academicYear: _selectedAcademicYear,
        );
        _saveData();
        _clearForm();
      });
    }
  }
  void _deleteStudent(int index) {
    setState(() {
      students.removeAt(index);
      _saveData();
});
  }
```

```
void _clearForm() {
  _formKey.currentState!.reset();
  _nameController.clear();
  _addressController.clear();
  _mobileController.clear();
  _emailController.clear();
  _prnController.clear();
  _selectedAcademicYear = 'First year'; // Reset selected academic year
}
void _showStudentDetails(int index) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Student Details'),
        content: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text('Name: ${students[index].name}'),
            Text('Address: ${students[index].address}'),
```

```
            Text('Mobile Number: ${students[index].mobileNumber}'),
            Text('Email: ${students[index].email}'),
            Text('PRN: ${students[index].prn}'),
            Text('Academic Year: ${students[index].academicYear}'),
          ],
        ),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: Text('Close'),
          ),
        ],
      );
    },
  );
}
@override
Widget build(BuildContext context) {
```

```
return Scaffold(
  appBar: AppBar(
    title: Text('Student Management App'),
    backgroundColor: Colors.blueGrey,
    actions: [
      PopupMenuButton<String>(
        onSelected: (value) {
          switch (value) {
            case 'Add':
              _addStudent();
              break;
            case 'View All':
            // Implement view all functionality
              break;
          }
        },
        itemBuilder: (BuildContext context) {
          return {'Add', 'View All'}.map((String choice) {
            return PopupMenuItem<String>(
              value: choice,
```

```
            child: Text(choice),
          );
        }).toList();
      },
    ),
  ],
),
body: Padding(
  padding: const EdgeInsets.all(16.0),
  child: Form(
    key: _formKey,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        TextFormField(
          controller: _nameController,
          decoration: InputDecoration(labelText: 'Name'),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter the name';
```

```
    }
    return null;
  },
),
TextFormField(
  controller: _addressController,
  decoration: InputDecoration(labelText: 'Address'),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter the address';
    }
    return null;
  },
),
TextFormField(
  controller: _mobileController,
  decoration: InputDecoration(labelText: 'Mobile Number'),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter the mobile number';
```

```
    }
    return null;
  },
),
TextFormField(
  controller: _emailController,
  decoration: InputDecoration(labelText: 'Email'),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter the email';
    } else if (!RegExp(
        r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b')
        .hasMatch(value)) {
      return 'Invalid email address';
    }
    return null;
  },
),
TextFormField(
  controller: _prnController,
```

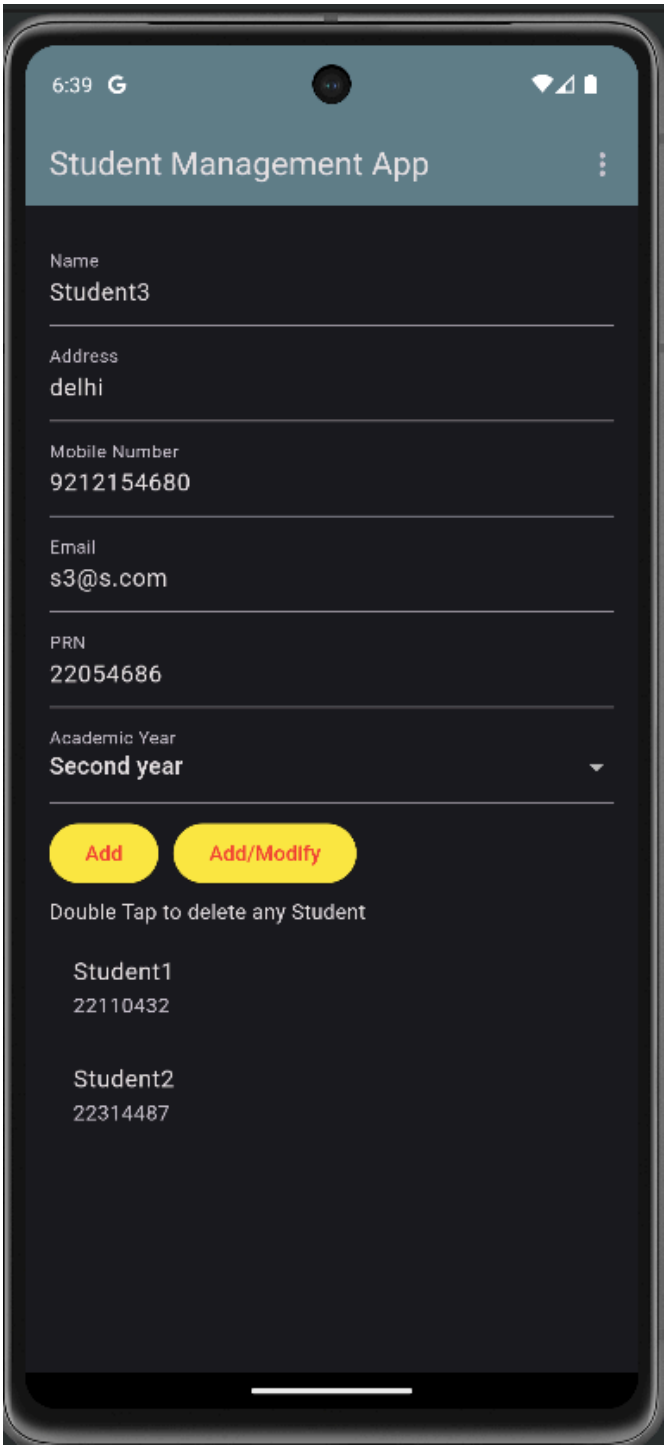
```
  decoration: InputDecoration(labelText: 'PRN'),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter the PRN';
    }
    return null;
  },
),
DropdownButtonFormField<String>(
  value: _selectedAcademicYear,
  onChanged: (String? newValue) {
    setState(() {
      _selectedAcademicYear = newValue!;
    });
  },
  items: <String>['First year', 'Second year', 'Third year', 'Fourth year']
      .map<DropdownMenuItem<String>>((String value) {
    return DropdownMenuItem<String>(
      value: value,
      child: Text(value),
```

```
    );
  }).toList(),
  decoration: InputDecoration(labelText: 'Academic Year'),
),
SizedBox(height: 10),
Row(
  children: [
    ElevatedButton(
      onPressed: _addStudent,
      child: Text('Add'),
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.yellow, // Change button background color to yellow
        foregroundColor: Colors.red,
      ),
    ),
    SizedBox(width: 10),
    ElevatedButton(
      onPressed: () {
        int selectedIndex = students.indexWhere(
              (student) => student.name == _nameController.text);
```

```
        if (selectedIndex != -1) {
          _modifyStudent(selectedIndex);
        } else {
          _addStudent();
        }
      },
      child: Text('Add/Modify'),
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.yellow,
        foregroundColor: Colors.red,
      ),
    ),
  ],
),
SizedBox(height: 5),
Text('Double Tap to delete any Student'),
SizedBox(height: 5),
Expanded(
  child: ListView.builder(
    itemCount: students.length,
```

```
                itemBuilder: (context, index) {
                  return GestureDetector(
                    onDoubleTap: () => _deleteStudent(index),/*to delete a student*/
                    child: ListTile(
                      title: Text(students[index].name),
                      subtitle: Text(students[index].prn),
                      onTap: () => _showStudentDetails(index),
                    ),
                  );
                },
              ),
            ),
          ],
        ),
      ),
    ),
  );
 }
 getApplicationDocumentsDirectory() {}
}
```

OUTPUT:


6:39
Student Management App
Name
Student3
Address
delhi
Mobile Number
9212154680
Email
s3@s.com
PRN
22054686
Academic Year
Second year
Add
Add/Modify
Double Tap to delete any Student
Student1
22110432
Student2
22314487