# DRIVING SCHOOL

Annit Thomas Mannathu

22PMC114

OVERVIEW

The platform allows users to create accounts and log in securely. The driving school aims to provide an interactive and educational platform for individuals learning to drive or looking to enhance their driving skills. The platform will be accessible to users anytime and anywhere, allowing them to learn at their own pace and convenience. This accessibility eliminates the need for physical attendance at driving schools, making it a cost-effective and time-efficient solution.
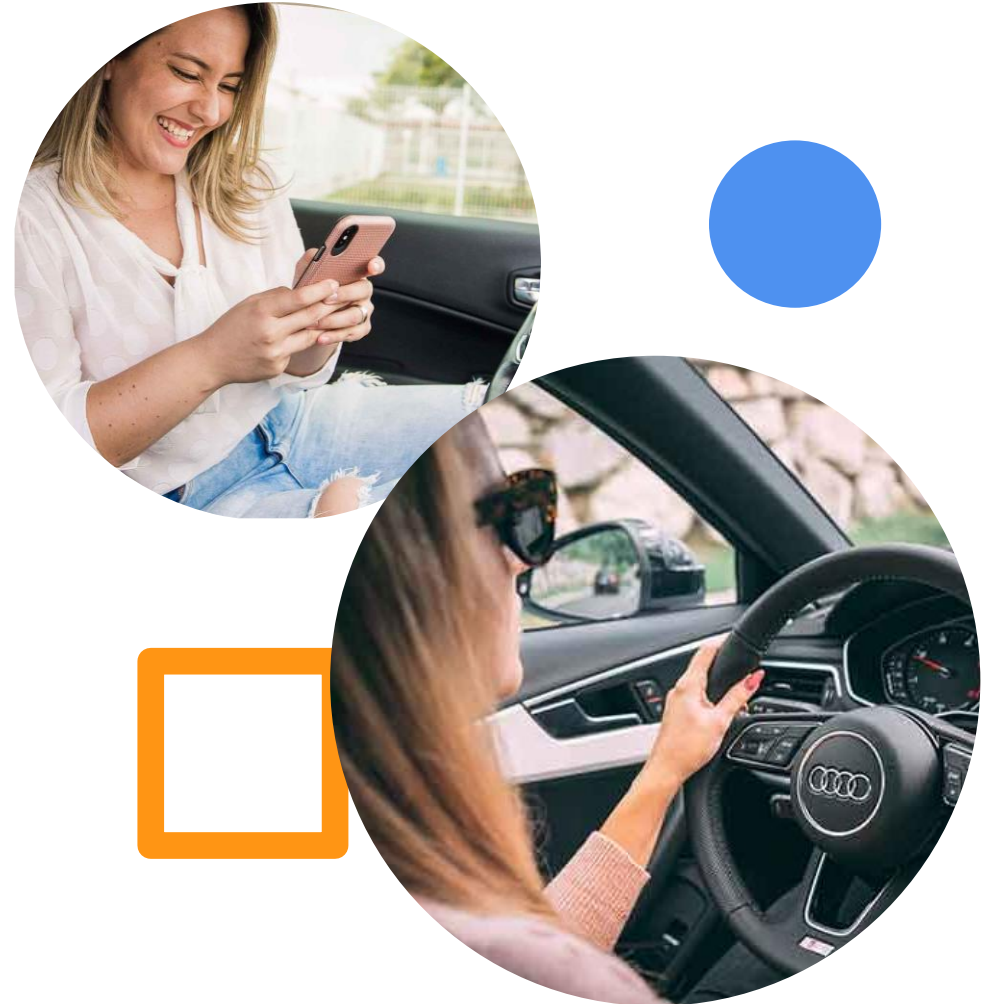
PROJECT REQUIREMENTS

1.User Registration and Authentication:

- Users should be able to create an account with relevant information (name, contact details, etc.).
- Implement secure authentication mechanisms for user login and password management.

2.Time Slot Management:

- Implement an administrative interface for driving school administrators to manage time slots.
- Admins should be able to add, modify, and remove time slots based on instructor availability.
- Time slots should have details such as date, time, instructor name, and available seats.

3. Booking System:

       Users should be able to view available time slots and book their preferred slot.

4. Security and Data Management:

- Implement security measures to protect user data and prevent unauthorized access.
- Safely store and manage user information, booking details, and transaction history.

5. Instructor can login only after the confirmation by admin.

6. Users can choose their tutor and they can select timing.

7. The system shall provide an option to add slot for instructors.

FEATURES AND HIGHLIGHTS OF YOUR PROJECT

1.User Registration and Authentication:
1. Users can create accounts and log in securely.
2. User authentication ensures data privacy and personalization of the booking experience.

2. Users should be able to create an account with relevant information (name, contact details, etc.).

3. Time Slot Management:
1. Admin interface allows driving school administrators to manage time slots.
2. Admins can add, modify and remove time slots based on instructor availability.

4. Booking System:
1. Users can view available time slots and book preferred slots.
2. Prevention of double booking ensures slot availability accuracy.

5. The system shall provide an option to add slot for instructors.
6. User Interfaces:
   User-friendly interfaces for students and instructors.

TECHNICAL ASPECTS

Django Framework:
It provides a robust foundation for developing web applications, offering features such as URL routing, database connectivity, authentication, and templating.

Python:
Python is used to implement the backend logic, handle data processing, and perform various system-level operations.

HTML/CSS/JavaScript:
The project utilizes front-end technologies such as HTML, CSS, and JavaScript to develop the user interface and enhance user interactions. These technologies are essential for creating responsive and visually appealing web pages.

Database Management System (DBMS):
The project utilizes a DBMS to store and manage data related to users, courses, content, assessments, and grades. Popular choices for DBMS in Django projects include PostgreSQL, MySQL, and SQLite.

THIRD PARTY LIBRARIES

Django REST Framework:
This library is used to build APIs for enabling seamless communication between the front-end and backend.

Bootstrap:
Bootstrap is a popular CSS framework used for responsive web design and UI components, making it easier to create visually appealing and mobile-friendly interfaces.

Pillow
A powerful library for image processing and manipulation.

THE ARCHITECTURE

Models:
The models define the data structures and relationships within the LMS, including entities such as Users, Courses, Enrollments, Content, Assessments, and Grades.

View Models:
View models handle the business logic data flow between the models and templates. They handle user requests, retrieve data from the database, and prepare the data for rendering in the templates.
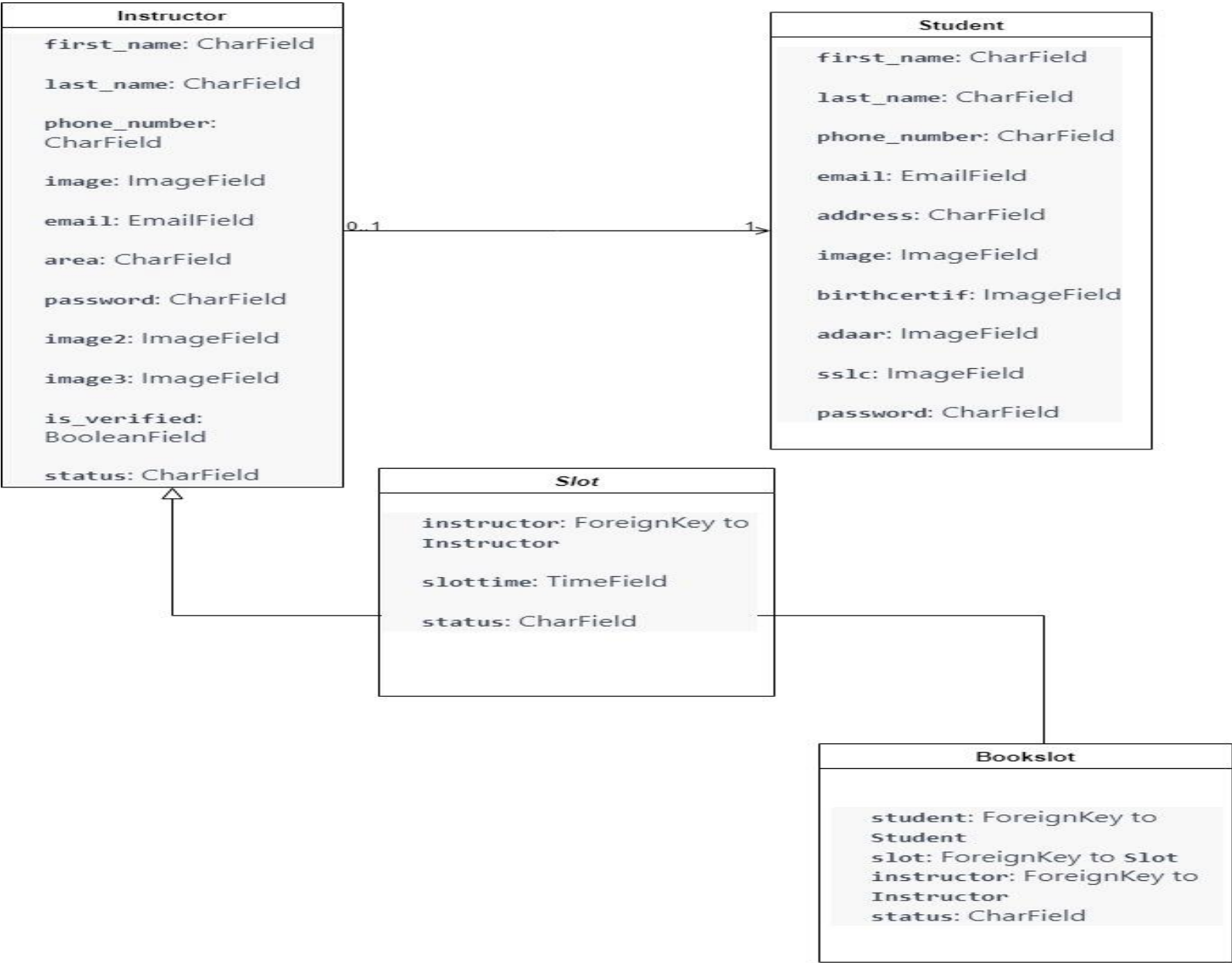
Templates:
Templates contain the HTML/CSS structure and presentation logic for rendering the user interface. They are responsible for displaying the data fetched from the views and rendering it in a user-friendly format.

Routers:
Routers handle the URL routing and map the incoming requests to the appropriate views or view models. They define the endpoints and HTTP methods for handling user requests.
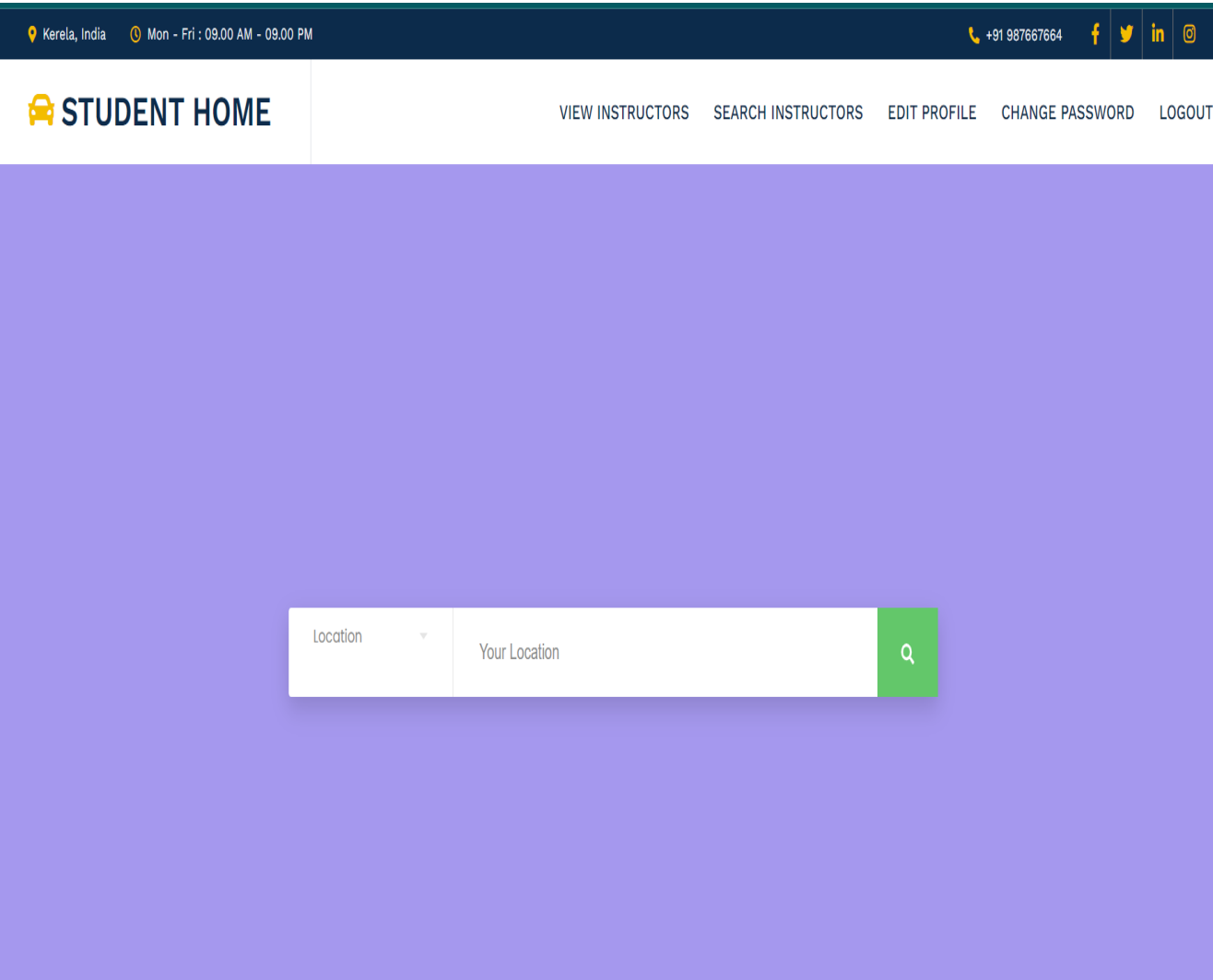
CLASS DIAGRAM

Instructor

first_name: CharField

last_name: CharField

phone_number: CharField

image: ImageField

email: EmailField

area: CharField

password: CharField

image2: ImageField

image3: ImageField

is_verified: BooleanField

status: CharField

Student

first_name: CharField

last_name: CharField

phone_number: CharField

email: EmailField

address: CharField

image: ImageField

birthcertif: ImageField

adaar: ImageField

sslc: ImageField

password: CharField

0..1 ⟶ 1

Slot

instructor: ForeignKey to Instructor

slottime: TimeField

status: CharField

Bookslot

student: ForeignKey to Student
slot: ForeignKey to Slot
instructor: ForeignKey to Instructor
status: CharField

CHALLENGES FACED DURING THE DEVELOPMENT

1. Complex Booking Rules and Availability Constraints.

2. User Experience and Interface Design.

3. Testing and Quality Assurance.

4. Handling Concurrent Booking Requests and Preventing Conflicts.

5. Performance Optimization.

IMPORTANT SCREENSHOTS WITH EXPLANATION

SEARCHING INSTRUCTOR

To enable searching for instructors based on location, you can implement a feature that allows students to find instructors within their desired locality.

SLOT ADDING

The slot adding page by the admin is a feature that allows the admin to add new slots for driving lessons to the system.

BOOKING SLOT

To create a student slot booking page, you'll need to design a web interface that allows students to view available slots and book appointments with instructors.

FUTURE ENHANCEMENTS

1.Payment Integration.

2. In-app Messaging System.

3.Waitlist Management.

4.Notifications.

CONCLUSION

In conclusion, the driving school mini project successfully addresses the key requirements of a modern driving school management system. It provides a user-friendly interface for both instructors, admin and students streamlining various processes and enhancing the overall efficiency of the driving school. Students can easily register for driving lessons and select the most convenient option for their schedule. The project also emphasizes security and privacy, implementing measures to protect personal information and ensure data integrity. User authentication and secure data storage mechanisms are in place to maintain confidentiality and build trust among learners.

Overall, the driving school mini project offers a comprehensive and efficient solution for managing various aspects of a driving school. It improves the user experience for both instructors and students, streamlines processes, enhances communication, and facilitates better organization and monitoring of driving lessons.

REFERENCES

https://nevonprojects.com/motor-driving-school-system/

https://www.academia.edu/38538776/DRIVING_SCHOOL_MONITORING_SYSTEM

https://www.studentprojects.live/studentprojectreport/online-driving-school/

Thank you

Presentation Title