ST. TERESA'S COLLEGE(AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

A Truck Service App

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER APPLICATIONS [TRIPLE MAIN]

Submitted By:

DEVIKA B R

B.Sc. Computer Applications [Triple Main]

SB22CA009

Under the guidance of

Ms. MARY SONA N.X

Assistant Professor

DEPARTMENT OF COMPUTER APPLICATION

(2022-2025)

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled **"TRUCK CONNECT APP"** is a bona fide record of the work done by **DEVIKA B R** (SB22CA009) during the year 2022– 2025 and submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Applications (Triple Main) under Mahatma Gandhi University, Kottayam.

ST. TERESA'S COLLEGE DEPT. OF COMPUTERS ERNAKULAM

17/03/2025

Head of the Department

Internal Examiner: MARY SONA N.X

Date: 17/3/25

External Examiner: Dr. Reseena M.N.

Date: 17/3/25

DECLARATION

I, DEVIKA B R (SB22CA009), B.Sc. Computer Applications [Triple Main] student of St. Teresa's College (Autonomous), Ernakulam, hereby declares that the project **'TRUCK CONNECT'** submitted for **Bachelor's Degree in Computer Application** is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

Date: 17.03.2025

Place:Ernakulam

DEVIKA B R

ACKNOWLEDGEMENT

I would like to convey my heartfelt gratitude to **Rev. Dr. Sr. Vinitha (CSST) Manager, Director Rev. Sr. Emeline (CSST) and Principal Dr. Alphonsa Vijaya Joseph** for providing me with this wonderful opportunity to work on a project with the topic Comparative Analysis of Machine Learning Algorithms and Interface Implementation. I would like to express my profound gratitude to the Head of the Department of Computer Applications Ms. Sheeba Emmanuel and **my project guide Ms. MARY Sona N.X** and all other faculty of the department for their contributions to the completion of my project. The completion of the project would not have been possible without their help and insights.Finally, I take this opportunity to Thank all those who have directly or indirectly helped me with my project.

DEVIKA B R

ABSTRACT

TRUCK CONNECT is a mobile app that connects users with truck drivers for local and long-distance transport, similar to Uber. Users can book trucks based on their transportation needs, check fare estimates, and make secure payments. Truck drivers can accept bookings, manage trips, and optimize their schedules efficiently. The app provides a simple and reliable platform for hassle-free truck services, making goods transportation more accessible and efficient for both businesses and individuals. By streamlining the process, **TRUCK CONNECT** helps reduce delays, improve service availability, and enhance overall logistics management. This report details the design, implementation, and potential future scope of TRUCK CONNECT.

CONTENTS

1. INTRODUCTION

BACKGROUND

Booking trucks for transportation is often difficult, time-consuming, and unreliable. **TRUCK CONNECT** is a digital platform designed to simplify this process, making truck services more accessible and efficient.

PROBLEM STATEMENT

Traditional truck booking methods involve delays, high costs, and uncertainty in availability. Users struggle to find suitable trucks, while drivers face difficulties in getting customers. **TRUCK CONNECT** aims to bridge this gap by providing a seamless connection between users and truck drivers.

OBJECTIVE

The objective of **TRUCK CONNECT** is to offer a simple and efficient mobile app for booking trucks. Users can check availability, get fare estimates, and make secure payments, while truck drivers can manage trips easily. The goal is to improve accessibility, reliability, and efficiency in truck services.

2.LITERATURE REVIEW

The logistics and transportation industry has undergone significant digital transformation with the advent of mobile applications. Traditional truck rental and booking systems often rely on manual processes, leading to inefficiencies, high costs, and delays. The Truck App aims to bridge this gap by providing a platform where users can book trucks for personal use and rent them for exporting and importing goods. This literature review explores existing solutions, their challenges, and how Truck App fits into the evolving landscape of logistics technology.

EXISTING PLAN

- Uber Freight
- Blackbuck
- Freight Pop

GAPS IDENTIFIED

High Commission Fees: Many existing platforms charge high commission rates from drivers and users.

Limited Availability: Some platforms struggle to provide enough vehicles during peak demand.

Complex Interfaces: Some apps have complex user interfaces that are not user-friendly for individuals unfamiliar with logistics operations.

Trust and Security Issues: Concerns over fraud, hidden charges, and unreliable drivers exist in many rental and booking apps.

HOW TRUCK CONNECT FILLS THE GAP

The Truck Connect app is designed to overcome the challenges faced by existing truck booking and rental platforms. By focusing on user accessibility, affordability, and security, it provides a comprehensive solution that caters to both personal and business logistics need.

Reducing High Commission Fees

Challenge: Many truck booking platforms charge high commission fees, making it expensive for both truck owners and customers.

Truck Connect Solution: Introduces a low-cost commission model or a subscription-based system for truck owners to reduce operational costs.

Ensures transparent pricing with no hidden charges, making it more affordable for users.

Ensuring Truck Availability

Challenge: Some platforms struggle to provide enough vehicles, especially during peak demand.

Truck Connect Solution: Implements a dynamic allocation system that efficiently matches available trucks with customer requests in real time.

Expands the network by encouraging independent truck owners to register on the platform.

Simplifying the User Interface

Challenge: Many existing apps have complex interfaces that are difficult for non-experts to navigate.

Truck Connect Solution: Features an intuitive and user-friendly interface, making it easy for individuals and businesses to book trucks.

Enhancing Trust and Security

Challenge: Users often face fraud, hidden charges, or unreliable drivers on some platforms.

Truck Connect Solution: Implements a driver verification system, where only registered and background-checked drivers can accept bookings.

Introduces customer and driver ratings to maintain service quality.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Uber Freight: Uber Freight is a digital freight brokerage platform that connects shippers with truck drivers for seamless and transparent load booking, similar to how Uber connects riders with drivers.

Blackbuck: Blackbuck is an Indian logistics and trucking platform that connects shippers with truck owners and fleet operators, providing digital freight matching, fleet management, and financial services to streamline trucking operations.

Freight Pop: Freight POP is a cloud-based Transportation Management System (TMS) that helps businesses streamline shipping, rate shopping, tracking, and freight invoice management across multiple carriers and modes.

3.2 PROPOSED SYSTEM

1. User-Friendly Booking – Customers can easily book trucks through a mobile or web application, reducing manual efforts and saving time.

2. Truck Listing and Rental – Truck owners can list their vehicles for rent, enabling better utilization of resources.

3. Real-Time Availability – Users can check the availability of trucks in real time, ensuring faster service.

4. Secure Authentication – The system includes secure login and registration features to protect user data.

5. Admin Panel – The system will include an admin panel to monitor user activities, manage truck listings, and ensure smooth operations.

3.3 SYSTEM SPECIFICATION

After the analyst has collected all required information regarding the software to be developed, and has removed all completeness, inconsistency, and anomalies from specification, he starts to systematically organize the requirements in the form of an SRS document. The software developers refer to the SRS document to make sure that they developed exactly what the customer requires. The SRS document helps the maintenance engineers to understand the Functionality of the new system.

3.4 OPERATING SYSTEM

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released on October 5, 2021. It succeeded Windows 10 (2015) and is available for free any Windows 10 devices that meet the new Windows 11 system requirement. It also improved virtual desktops, gaming performance, and multitasking capabilities. However, it also has several disadvantages such as incompatible hardware, limited compatibility with older software, and fewer customization options.

3.5 LANGUAGE OR SOFTWARE PACKAGE

We are using Java for Android to develop the front-end of the app, providing a robust and flexible mobile interface. For the backend, we've chosen JSP (Java Server Pages) to handle dynamic content and server-side logic, while MySQL serves as the relational database to efficiently manage and store our data. MySQL is a widely used open-source Relational Database Management System (RDBMS) that utilizes Structured Query Language (SQL) for database operations. Developed by MySQL AB and now owned by Oracle Corporation, MySQL is known for its reliability, speed, and ease of use. It is a key component in many web applications, forming the backbone of popular websites and services. Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent methods for building WEB based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. This tutorial will teach you how to use Java Server Pages to develop your web applications in simple and easy steps. Android Studio is the official Integrated Development Environment (IDE) for Android app development, and it supports Java as one of the primary programming languages. Here's a step-by-step guide on how to use Java in Android Studio to build your first Android app.

3.6 HARDWARE AND SOFTWARE SPECIFICATION

- **Front End:** Android (Java)
- **Back End**: Jsp
- **Database Management System**: MySQL
- **Software used**: Wamp studio, Net Beans, SQLYog
- **Operating System**: Microsoft Windows 10 or above, Browser: Google Chrome

4. SYSTEM DESIGN

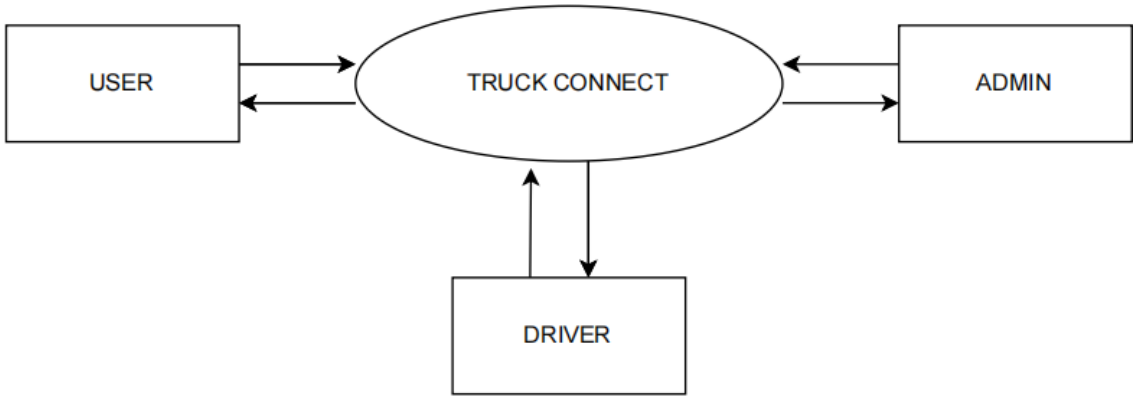
INTRODUCTION

System design is an interactive process through which requirements are transmitted to a "blueprint" for constructing the software initially and the blueprint depicts a holistic view of software whose design is represented at a high-level abstraction. A level that can be directly traced to specific data, functional and behavioural requirements. As design interaction occurs subsequent refinement led to design representation at a much lower level of abstraction. System design is a creative art of inventing and developing input, databases, offline files, methods, and procedures, for processing data to get meaningful full output that satisfies the organization objectives. Through the design phase consideration to the human factor, that is inputs to the users will have on the system. Some of the main factors that must be noted using the design of the system are:

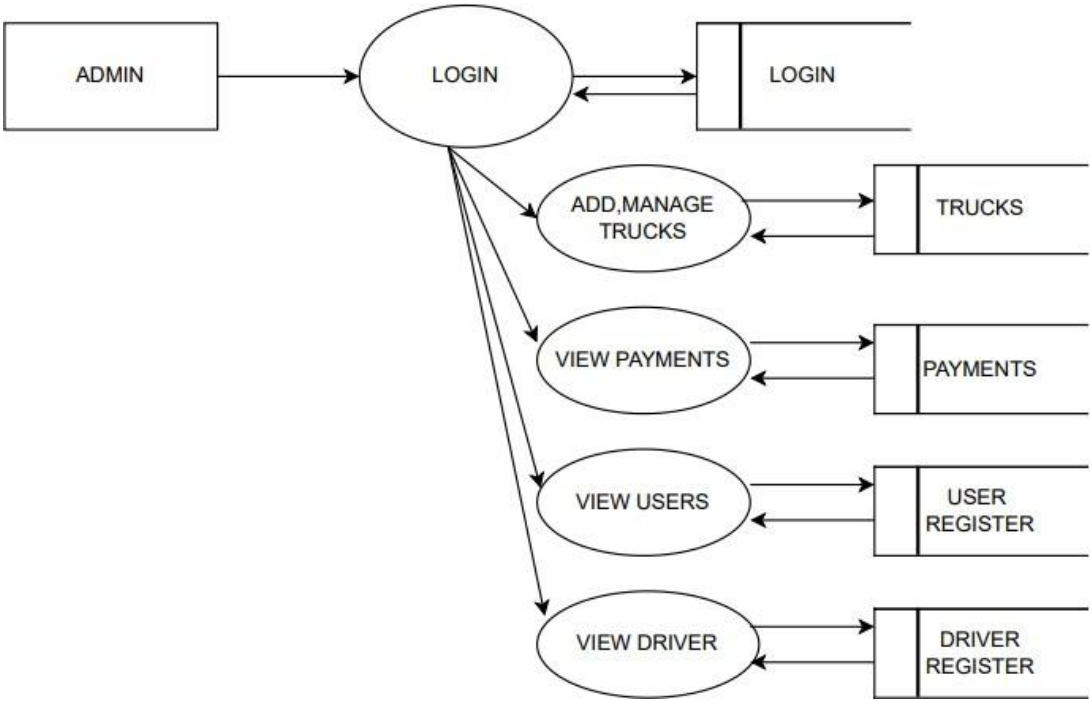
- Practicality: System must be capable of being operated over a long period of time and must have ease of use.
- Efficiency: Make better use of available resources. Efficiency involves accuracy, timeliness, and comprehensive system output.
- Cost: Aim of minimum cost and better results.
- Security: Ensure physical security of data**.**

4.2 DATA FLOW DIAGRAM

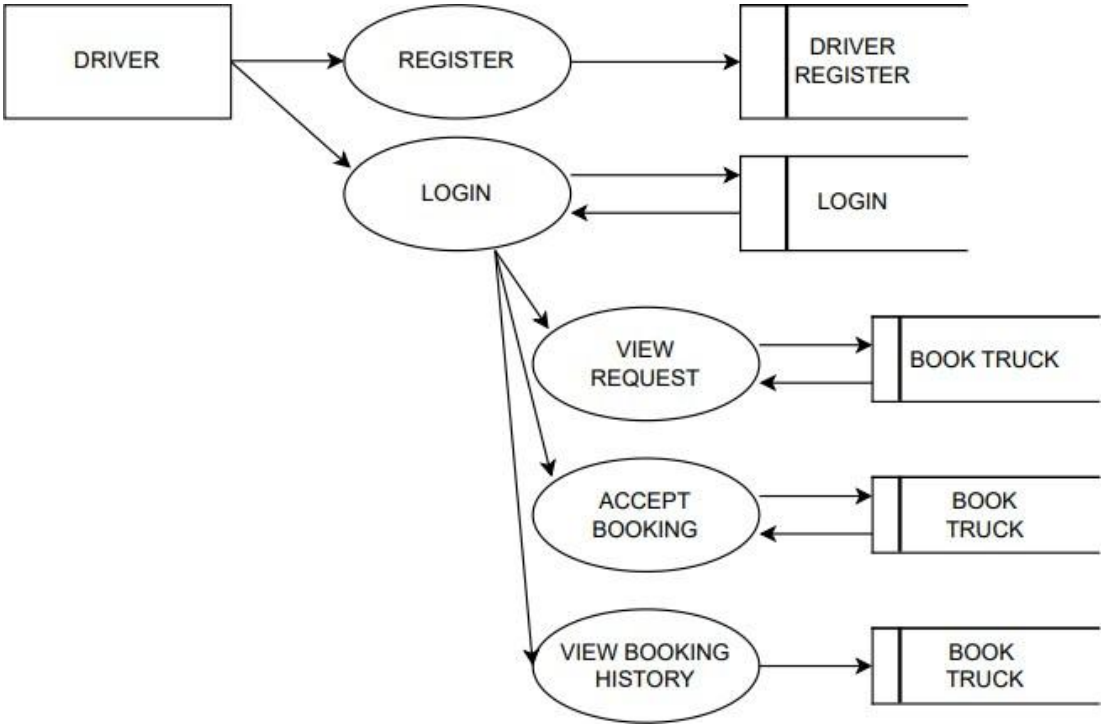
Level 0



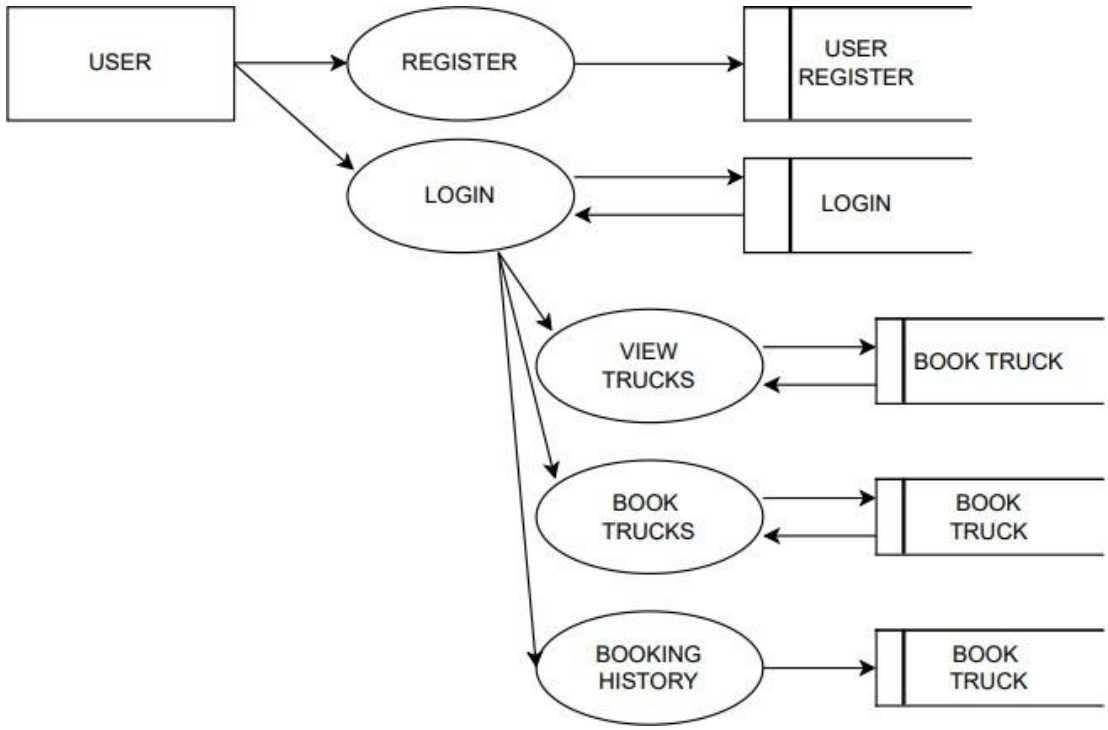
Level 1



Level 2



Level 3



4.3 DATABASE DIAGRAM

Database design, A most important part of the system design phase. In a database environment, data available are used by several users instead of each program managing its own data, authorized users share data across application with the database software managing the data as an entity. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive, and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of mana-gement information system and is available virtually for every computer system.

User Register

Field	Type	Null	Default	Comments
user_id	int(255)	No		
user_name	varchar(500)	Yes	NULL	
user_mobile	varchar(500)	Yes	NULL	
user_email	varchar(500)	Yes	NULL	
user_date_of_join	Timestamp	Yes	CURRENT_TIMESTAMP	

Driver Register

Field	Type	Null	Default	Comments
dr_id	int(255)	No		
dr_Name	varchar(255)	Yes	NULL	
dr_liscenseNumber	varchar(255)	Yes	NULL	
dr_contactNumber	varchar(255)	Yes	NULL	
dr_email	varchar(255)	Yes	NULL	
dr_experience	varchar(250)	Yes	NULL	
dr_status	varchar(500)	Yes	active	

Book Truck

Field	Type	Null	Default	Comments
book_id	int(255)	No		
truck_id	int(255)	Yes	NULL	
user_id	int(255)	Yes	NULL	
driver_id	int(255)	Yes	NULL	
book_date	varchar(500)	Yes	NULL	
book_time	varchar(500)	Yes	NULL	
book_destination	varchar(500)	Yes	NULL	
book_kilometers	varchar(500)	Yes	NULL	
book_amount	varchar(500)	Yes	NULL	
book_paymentStatus	varchar(500)	Yes	pending	
book_status	varchar(500)	Yes	pending	

Trucks

Field	Type	Null	Default	Comments
truck_id	int(255)	No		
truck_name	varchar(500)	Yes	NULL	
truck_number	varchar(500)	Yes	NULL	
truck_capacity	varchar(500)	Yes	NULL	
truck_type	varchar(500)	Yes	NULL	
truck_status	varchar(500)	Yes	active	
truck_image	Longblob	Yes	NULL	
truck_rating	Double	Yes	1.5	
truck_driver	int(255)	Yes	2	

Charge

Field	Type	Null	Default	Comments
charge_id	int(255)	No		
truck_type	varchar(500)	Yes	NULL	
charge	varchar(500)	Yes	NULL	

Login

Field	Type	Null	Default	Comments
login_id	int(50)	No		
registration_id	int(50)	Yes	NULL	
email	varchar(100)	Yes	NULL	
password	varchar(100)	Yes	NULL	
type	varchar(100)	Yes	NULL	
status	varchar(100)	Yes	NULL	

5. SYSTEM TESTING AND IMPLEMENTATION

5.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of the specification, design, and coding. System testing makes a logical assumption that all parts of the system are correct; the goal will be successfully achieved. Implementation allows the users to take over its operation for use and evaluation. Maintenance changes the existing system, enhancement adds features to the existing system, and development replaces the existing system.

5.2 SYSTEM IMPLEMENTATION

Implementation phase is the phase, which involves the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. Implementation is the stage of the project, where the theoretical design is turned into a working system. At this stage the main workload, the greatest up heal and the major impact on existing practices shift to the user department. If the implementation stage is not planned and controllers carefully, it can cause chaos. The implementation stage is a system project. It involves careful planning, investigation of the current system and its constraints on the implementation, design methods to achieve the changeover procedures, and evaluation of change over methods. The implementation plan consists of the following steps:

• Testing the developed system with the sample data.

• Detection and correction of errors.

• Making necessary changes in the system.

• Training and involvement of user personnel.

• Installation of software utilities.

5.3 PROJECT SCOPE

In future, many enhancements can be made some of them include:

• Stay updated with emerging web technologies for potential upgrades.

• We can upgrade our application to manage the sports activities beyond a specific locality.

• Visitors and participants can register their complaints to the authorities

6. FEATURES AND FUNCTIONALITIES

User Features

Easy Registration and login - Easy to Sign up and login using email id.

Truck Booking - Users can book trucks for personal use like moving furniture, shifting houses etc.

Booking History - Users can check past activities and trip details.

Truck Owner Features

Truck listing - Truck owners can register their vehicle with details like capacity, type.

Manage Booking - Owners get real time notification when someone books their truck. They can accept or decline requests.

Admin Features

User and Truck Management - Admin can verify, approve, or, suspend user and truck owner.

Booking and Payment - Admin can track transactions, ensure secure payment.

7. FUTURE ENHANCEMENTS

1. Smarter, Fairer Pricing

Imagine if Truck Connect could automatically find you the best price based on real-time factors like traffic, fuel costs, and truck availability. No more guesswork—just fair pricing every time.

2. Faster, More Efficient Routes

Ever wished your truck delivery could take the smartest route without getting stuck in traffic? The app could use AI to guide drivers on the best roads, saving time and fuel.

3. Sharing a Truck, Saving Money

Not everyone needs an entire truck, right? What if you could split the space with someone else going the same way? Load-sharing could make it cheaper for users and more profitable for truck owners.

4. Keeping an Eye on Trucks in Real Time

Truck owners could install smart sensors in their vehicles to monitor fuel levels, engine health, and even driver behaviour. This way, they'd know if a truck needs maintenance before it breaks down on the road.

5. Helping Truck Owners Get More Bookings

If a truck owner knew exactly where demand was high, they could move their trucks there in advance. The app could analyse data and suggest where their next job might be waiting.

6. Secure, No-Hassle Payments

Imagine finishing a trip and the payment happens automatically—no delays, no disputes. With blockchain-based smart contracts, payments could be locked in and released only when the job is done right.

7. Business-Friendly Bulk Bookings

Big businesses often need multiple trucks at once. The app could allow them to book an entire fleet in one go, making large-scale transport effortless.

8. Partnering with E-Commerce & Warehouses

What if online businesses and warehouses could instantly book trucks whenever they needed them? The app could integrate directly with their systems, making logistics smoother.

9. A Personal Truck Assistant

Need to book a truck while driving or busy? Just ask a voice assistant. "Hey, Truck Connect, find me a mini-truck for today at 4 PM," and it's done.

10. A Greener Way to Move Goods

Trucks contribute to pollution, but what if the app helped users choose electric trucks or fuel-efficient routes? Businesses could even get reports on their carbon footprint and make eco-friendly choices.

These upgrades would turn Truck Connect into more than just a booking app—it would become a smarter, more efficient, and even environmentally responsible way to move goods.

8. CONCLUSION

CONCLUSION

Right now, Truck Connect is like a helping hand for people who need trucks—whether for personal shifting or big business deliveries. It simplifies the process, making truck booking as easy as ordering a cab.

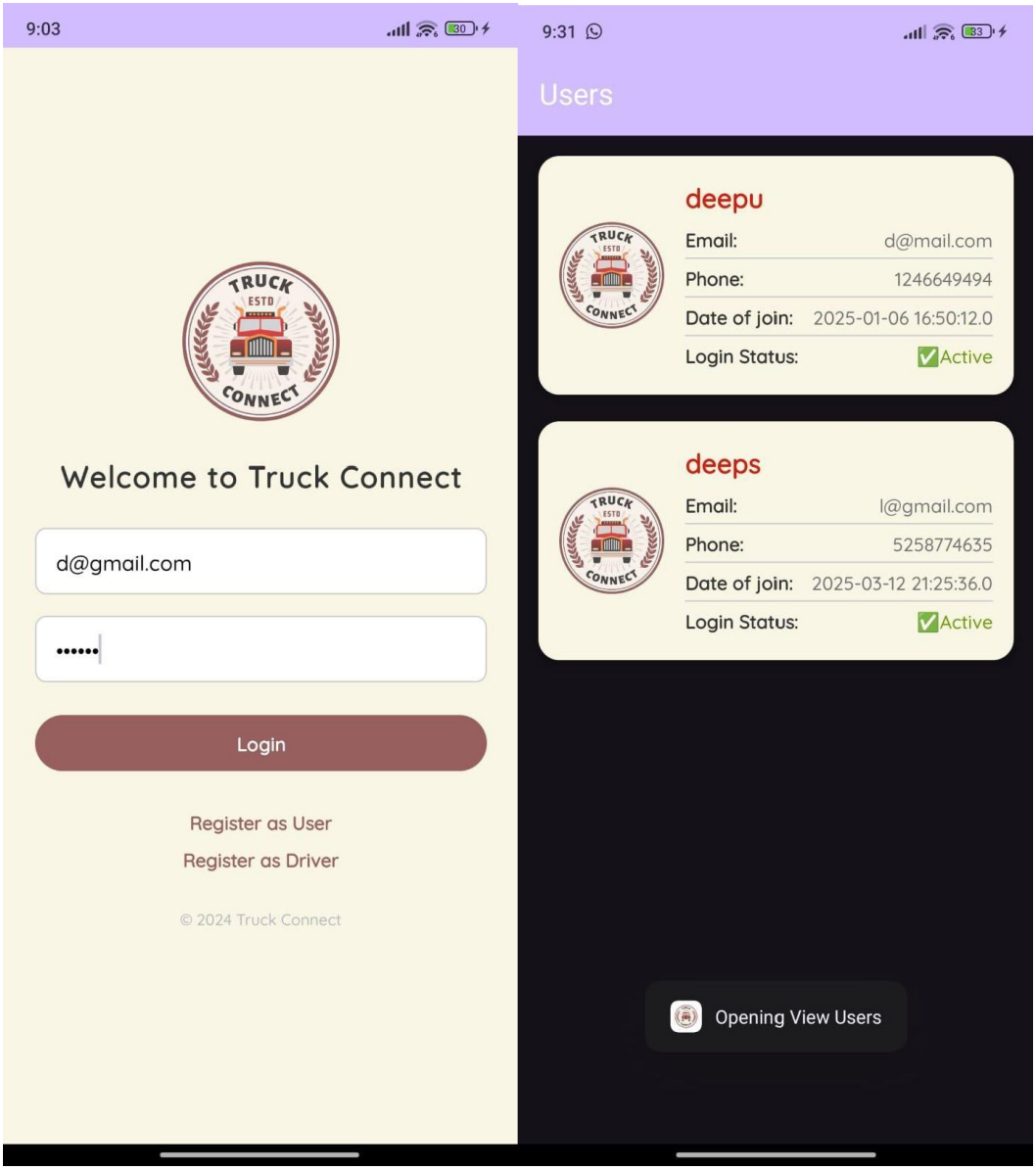
But the future? That's where things get really exciting. With AI-powered pricing, smarter routes, load-sharing, and real-time tracking, it won't just be an app—it'll be a full-fledged logistics assistant. Imagine a world where trucks are always in the right place at the right time, where businesses can move goods effortlessly, and where even small users can book a truck without stress.

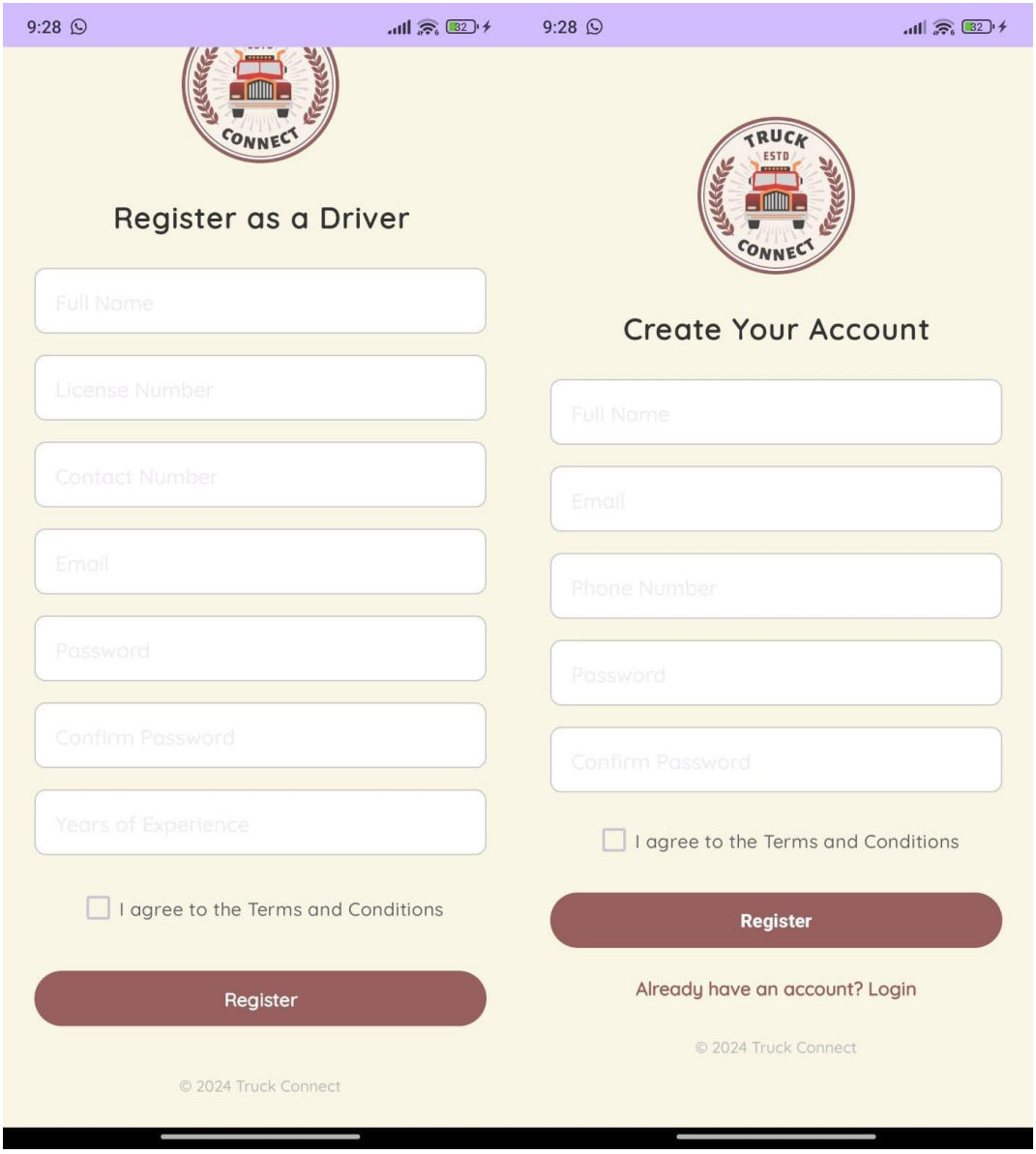
And the best part? Truck Connect can grow and evolve based on what users really need. Whether it's seamless payments, AI suggestions, or eco-friendly transport options, the possibilities are endless.

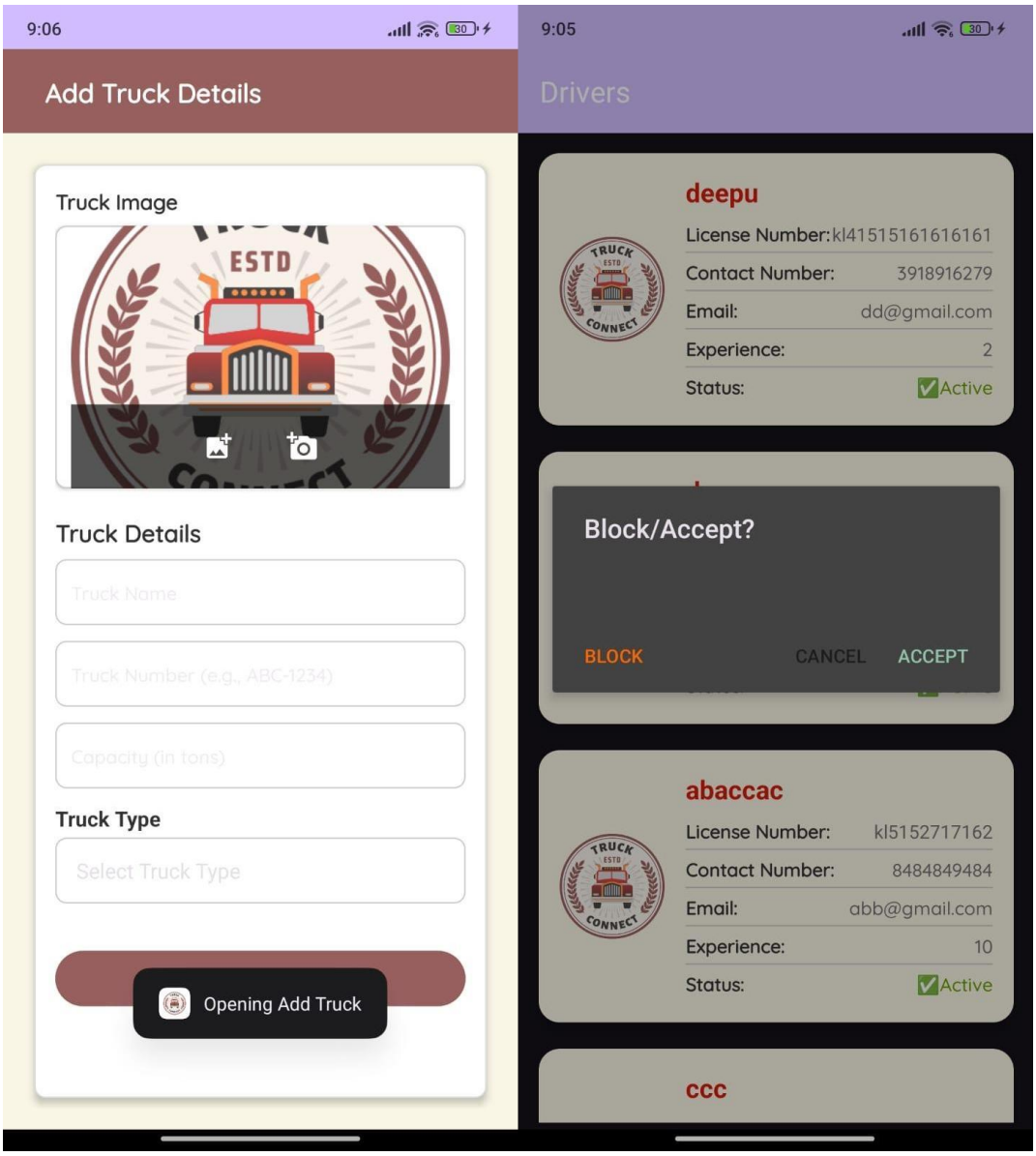
In short, Truck Connect is just getting started. The road ahead is full of upgrades, and each one will make truck booking smarter, smoother, and better for everyone

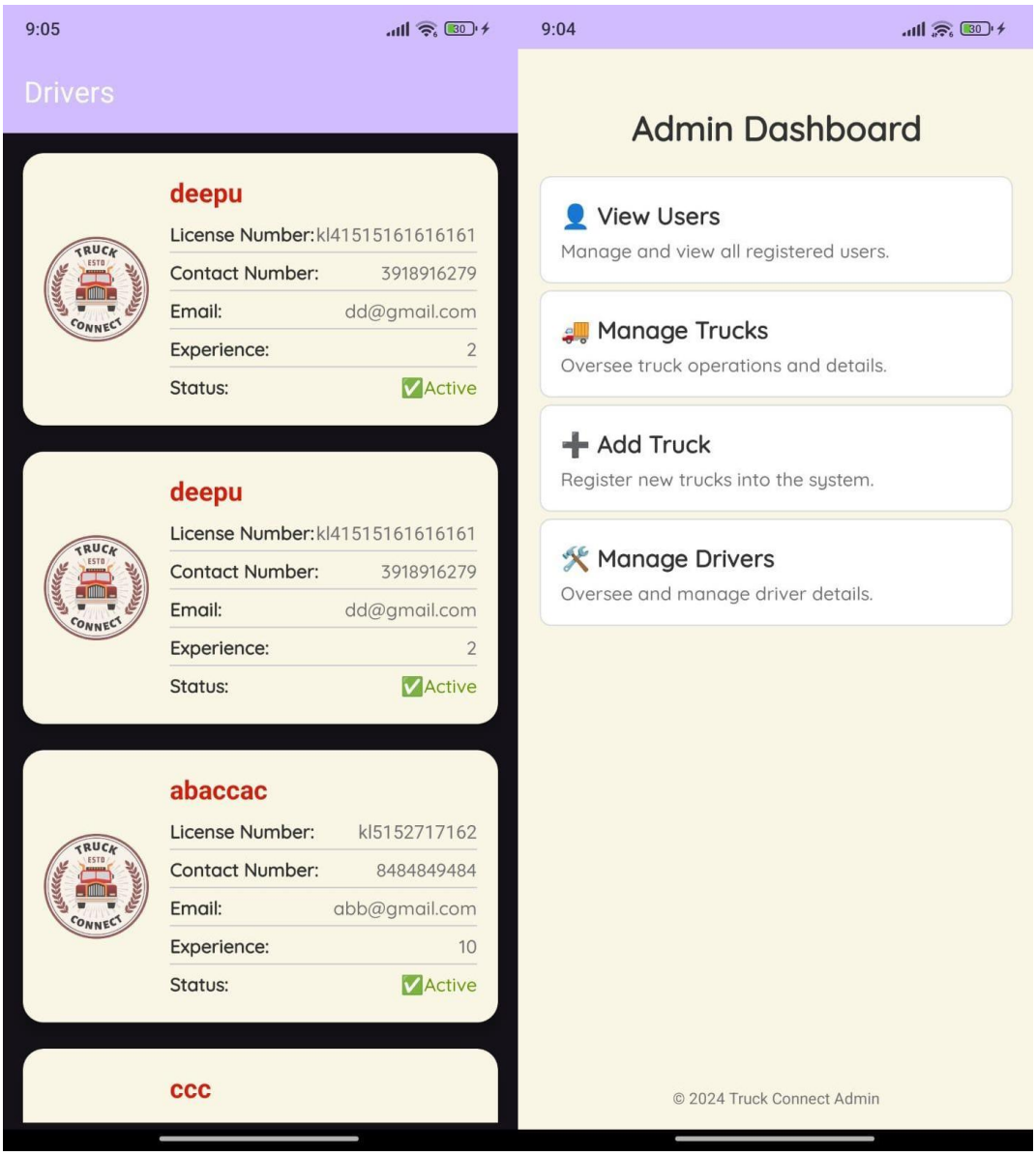
9.APPENDIX

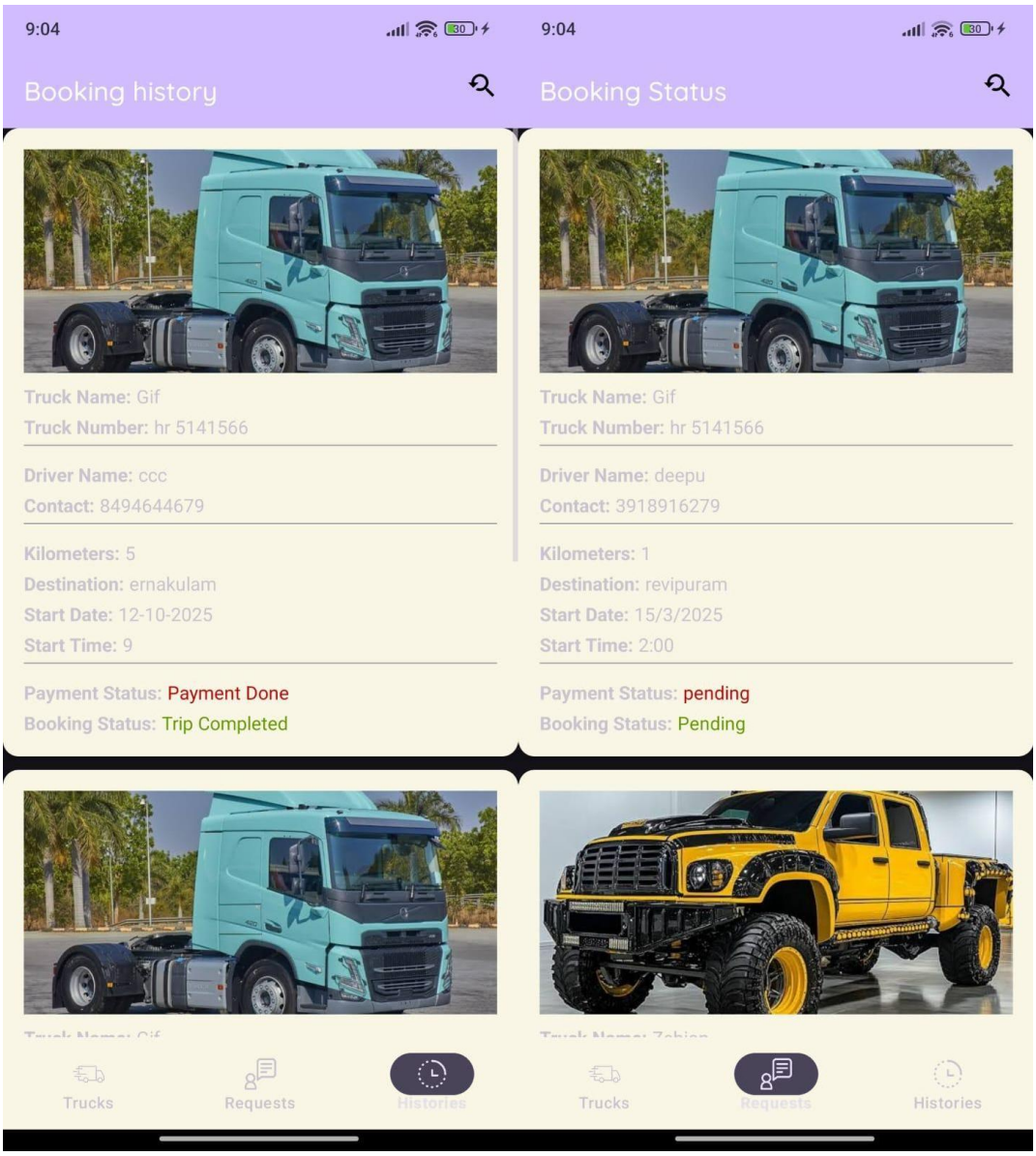
9.1 INPUT AND OUTPUT SCREEN

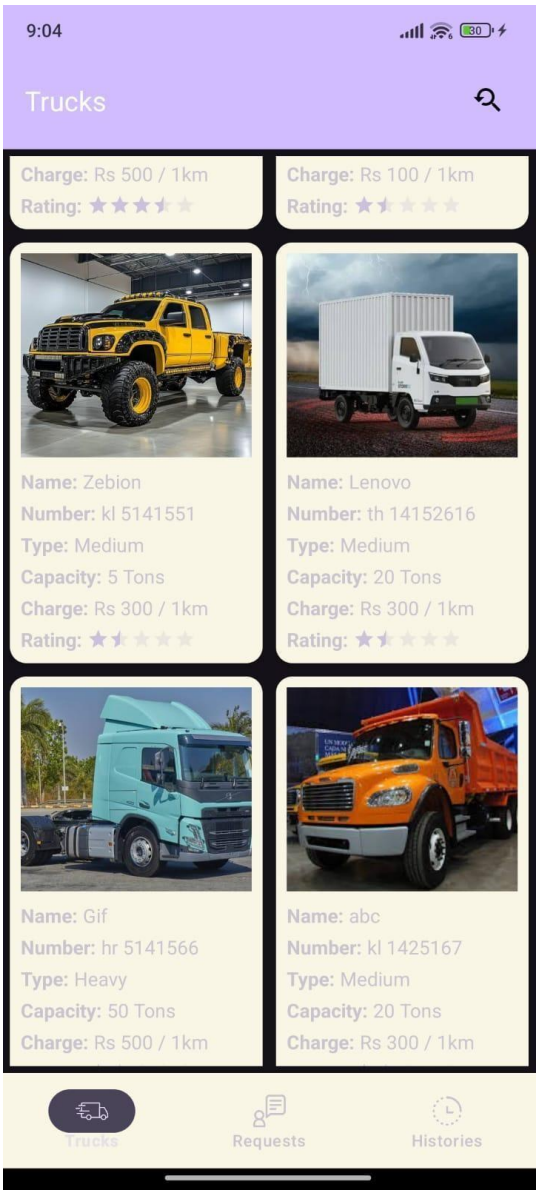



9:28
Register as a Driver
Full Name
License Number
Contact Number
Email
Password
Confirm Password
Years of Experience
I agree to the Terms and Conditions
Register
© 2024 Truck Connect
9:28
TRUCK
ESTD
CONNECT
Create Your Account
Full Name
Email
Phone Number
Password
Confirm Password
I agree to the Terms and Conditions
Register
Already have an account? Login
© 2024 Truck Connect


9:06
Add Truck Details
Truck Image
ESTD
Truck Details
Truck Name
Truck Number (e.g., ABC-1234)
Capacity (in tons)
Truck Type
Select Truck Type
Opening Add Truck
9:05
Drivers
deepu
License Number: kl41515161616161
Contact Number: 3918916279
Email: dd@gmail.com
Experience: 2
Status: Active
Block/Accept?
BLOCK
CANCEL
ACCEPT
abaccac
License Number: kl5152717162
Contact Number: 8484849484
Email: abb@gmail.com
Experience: 10
Status: Active
ccc


9:05
Drivers
deepu
License Number: kl41515161616161
Contact Number: 3918916279
Email: dd@gmail.com
Experience: 2
Status: ✅Active
deepu
License Number: kl41515161616161
Contact Number: 3918916279
Email: dd@gmail.com
Experience: 2
Status: ✅Active
abaccac
License Number: kl5152717162
Contact Number: 8484849484
Email: abb@gmail.com
Experience: 10
Status: ✅Active
ccc
9:04
Admin Dashboard
View Users
Manage and view all registered users.
Manage Trucks
Oversee truck operations and details.
Add Truck
Register new trucks into the system.
Manage Drivers
Oversee and manage driver details.
© 2024 Truck Connect Admin


9:04
Booking history
Truck Name: Gif
Truck Number: hr 5141566
Driver Name: ccc
Contact: 8494644679
Kilometers: 5
Destination: ernakulam
Start Date: 12-10-2025
Start Time: 9
Payment Status: Payment Done
Booking Status: Trip Completed
Trucks
Requests
Histories
9:04
Booking Status
Truck Name: Gif
Truck Number: hr 5141566
Driver Name: deepu
Contact: 3918916279
Kilometers: 1
Destination: revipuram
Start Date: 15/3/2025
Start Time: 2:00
Payment Status: pending
Booking Status: Pending
Trucks
Requests
Histories


9:04
Trucks
Charge: Rs 500 / 1km
Rating:
Charge: Rs 100 / 1km
Rating:
Name: Zebion
Number: kl 5141551
Type: Medium
Capacity: 5 Tons
Charge: Rs 300 / 1km
Rating:
Name: Lenovo
Number: th 14152616
Type: Medium
Capacity: 20 Tons
Charge: Rs 300 / 1km
Rating:
Name: Gif
Number: hr 5141566
Type: Heavy
Capacity: 50 Tons
Charge: Rs 500 / 1km
Name: abc
Number: kl 1425167
Type: Medium
Capacity: 20 Tons
Charge: Rs 300 / 1km
Trucks
Requests
Histories

9.2 SOURCE CODE

```
Package com.example.truckconn

ect.Authentication;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.truckconnect.Common.Utility;
import com.example.truckconnect.R;

import java.util.HashMap;
import java.util.Map;

public class RegistrationActivity extends AppCompatActivity {

    // UI Components
    private EditText fullNameInput, emailInput, phoneInput, passwordInput,
confirmPasswordInput;
    private CheckBox termsCheckbox;
    private Button registerButton;
    private TextView loginLink;

    @Override
    protected void onCreate(Bundle savedInstanceState)
```

```
{super.onCreate(savedInstanceState)
setContentView(R.layout.activity_registration);

        // Initialize UI Components
        fullNameInput = findViewById(R.id.fullNameInput);
        emailInput = findViewById(R.id.emailInput);
        phoneInput = findViewById(R.id.phoneInput);
        passwordInput = findViewById(R.id.passwordInput);
        confirmPasswordInput = findViewById(R.id.confirmPasswordInput);
        termsCheckbox = findViewById(R.id.termsCheckbox);
        registerButton = findViewById(R.id.registerButton);
        loginLink = findViewById(R.id.loginLink);

        // Register Button Click Listener
        registerButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                validateAndRegister();
            }
        });

        // Navigate to Login Page
        loginLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish(); // Close RegistrationActivity and go back to LoginActivity
            }
        });
    }

    // Validate all fields and call Register function
    private void validateAndRegister() {
        String fullName = fullNameInput.getText().toString().trim();
        String email = emailInput.getText().toString().trim();
        String phone = phoneInput.getText().toString().trim();
        String password = passwordInput.getText().toString().trim();
        String confirmPassword =
confirmPasswordInput.getText().toString().trim();
        boolean isTermsAccepted = termsCheckbox.isChecked();

        // Validation
```

```
        if (fullName.isEmpty()) {
            fullNameInput.setError("Full Name is required");
            fullNameInput.requestFocus();
            return;
        }

        if (email.isEmpty() ||
!android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            emailInput.setError("Valid Email is required");
            emailInput.requestFocus();
            return;
        }

        if (phone.isEmpty() || phone.length() != 10) {
            phoneInput.setError("Valid 10-digit Phone Number is required");
            phoneInput.requestFocus();
            return;
        }

        if (password.isEmpty()) {
            passwordInput.setError("Password is required");
            passwordInput.requestFocus();
            return;
        }

        if (!password.equals(confirmPassword)) {
            confirmPasswordInput.setError("Passwords do not match");
            confirmPasswordInput.requestFocus();
            return;
        }

        if (!isTermsAccepted) {
            Toast.makeText(this, "Please accept Terms & Conditions",
Toast.LENGTH_SHORT).show();
            return;
        }

        // If all fields are valid, call the Register functionC 6C
        Register(fullName, email, phone, password);
    }
  // Registration Logic
```

```
   private void Register(String Name, String email, String phone, String
password) {
      RequestQueue queue =
Volley.newRequestQueue(getApplicationContext());

      StringRequest request = new StringRequest(Request.Method.POST,
Utility.SERVERUrl, new Response.Listener<String>() {
         @Override
         public void onResponse(String response) {
            if (response.trim().equals("Already Exist")) {

               Toast.makeText(getApplicationContext(), " Already Exists",
Toast.LENGTH_SHORT).show();

            } else if (!response.trim().equals("failed")) {

               Toast.makeText(getApplicationContext(), "Success ..!",
Toast.LENGTH_SHORT).show();

               Intent intent = new Intent(getApplicationContext(),
LoginActivity.class);
               startActivity(intent);

            } else {
               Toast.makeText(getApplicationContext(), " Failed",
Toast.LENGTH_SHORT).show();
            }
         }
      }, new Response.ErrorListener() {
         @Override
         public void onErrorResponse(VolleyError error) {

            Toast.makeText(getApplicationContext(), "my error :" + error,
Toast.LENGTH_LONG).show();
            Log.i("My error", "" + error);
         }
      }) {
         @Override
         protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> map = new HashMap<String, String>();
            map.put("key", "userRegister");
```

```
                map.put("name", Name);
                map.put("email", email);
                map.put("phone",phone);
                map.put("password", password);

                return map;
            }
        };
        queue.add(request);

    }
}
```

10. BIBLIOGRAPHY

Reference

- https://www.java.com/
- https://developer.android.com/studio
- https://www.mysql.com/
- https://www.geeksforgeeks.org/introduction-to-jsp/