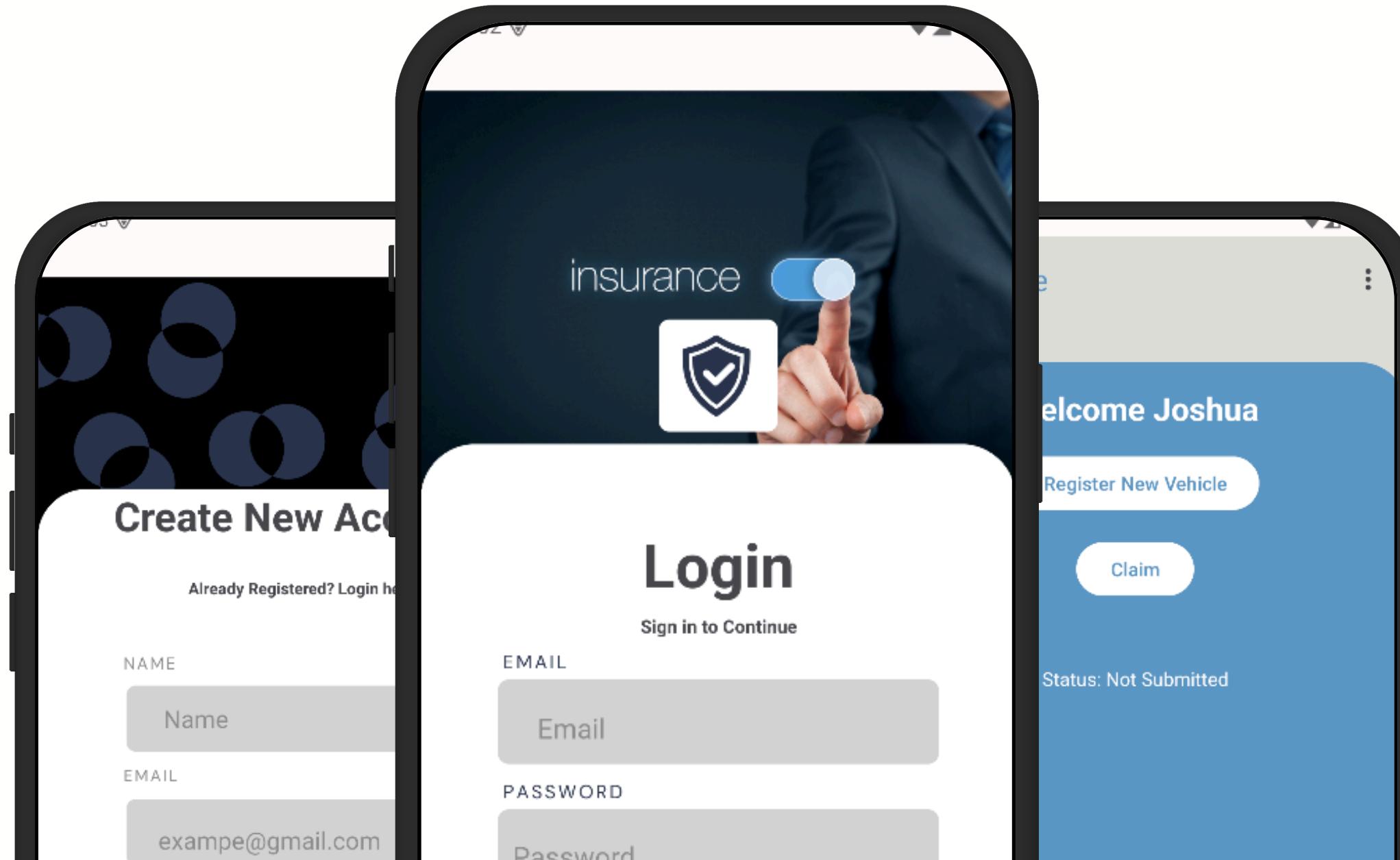


Vehicle Insurance Mobile App



Vehicle Insurance App

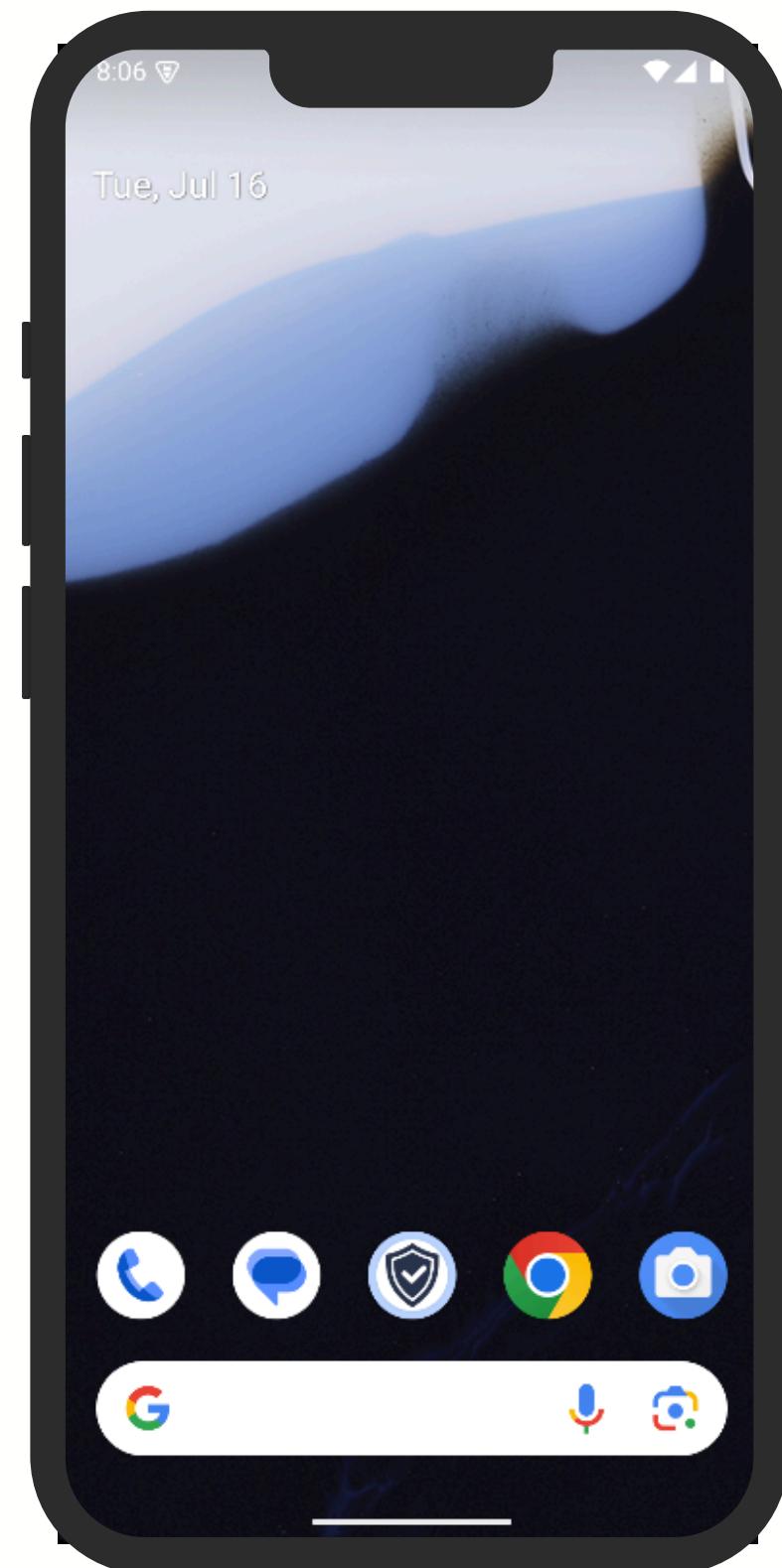
The Vehicle Insurance App aims to simplify the process of managing vehicle insurance claims and policies through a user-friendly mobile interface. Key features include secure user authentication, real-time claim status tracking, vehicle registration, and policy management.

This app caters to both users seeking efficient claim filing and insurance agents managing policy information.

NEXT →

Technologies used

- **Android Studio for frontend development**
- **Java and XML for UI design and logic**
- **Node.js and Express.js for backend services**
- **MongoDB for data storage and management**



Notch

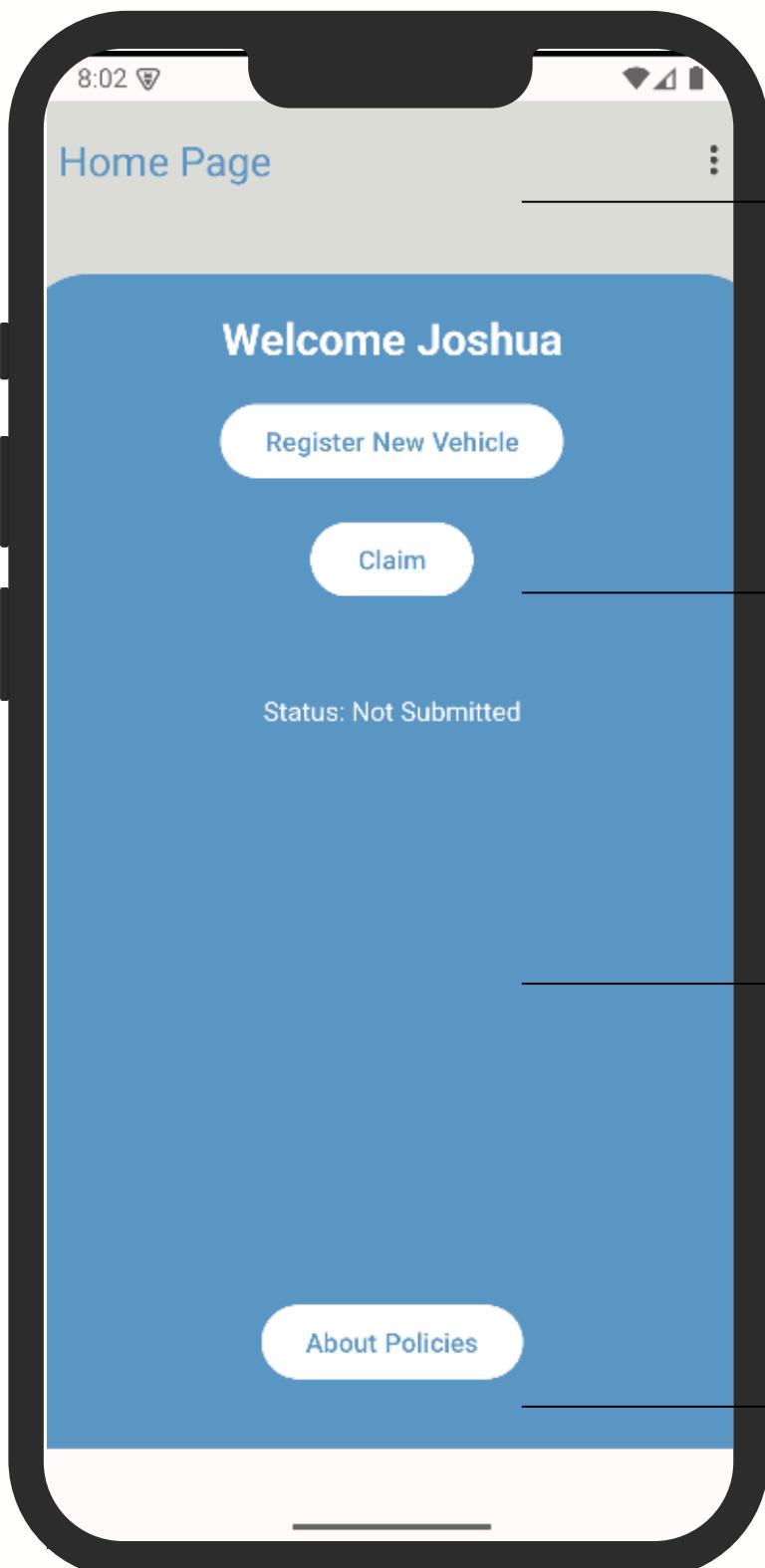
Features

Session Manager

- Saves the Login Details
- Saves the Vehicles Details
- Until you press logout. You wont be logout

Shared Preferences

- In this saved preferences the vehicle details will be saved, when the user logins it shows the latest details that are submitted.
- Your name will be displayed when you login with a welcome message.



● Menu

Menu contains like Profile, Settings, Help, About, Logout

● Register New Vehicle

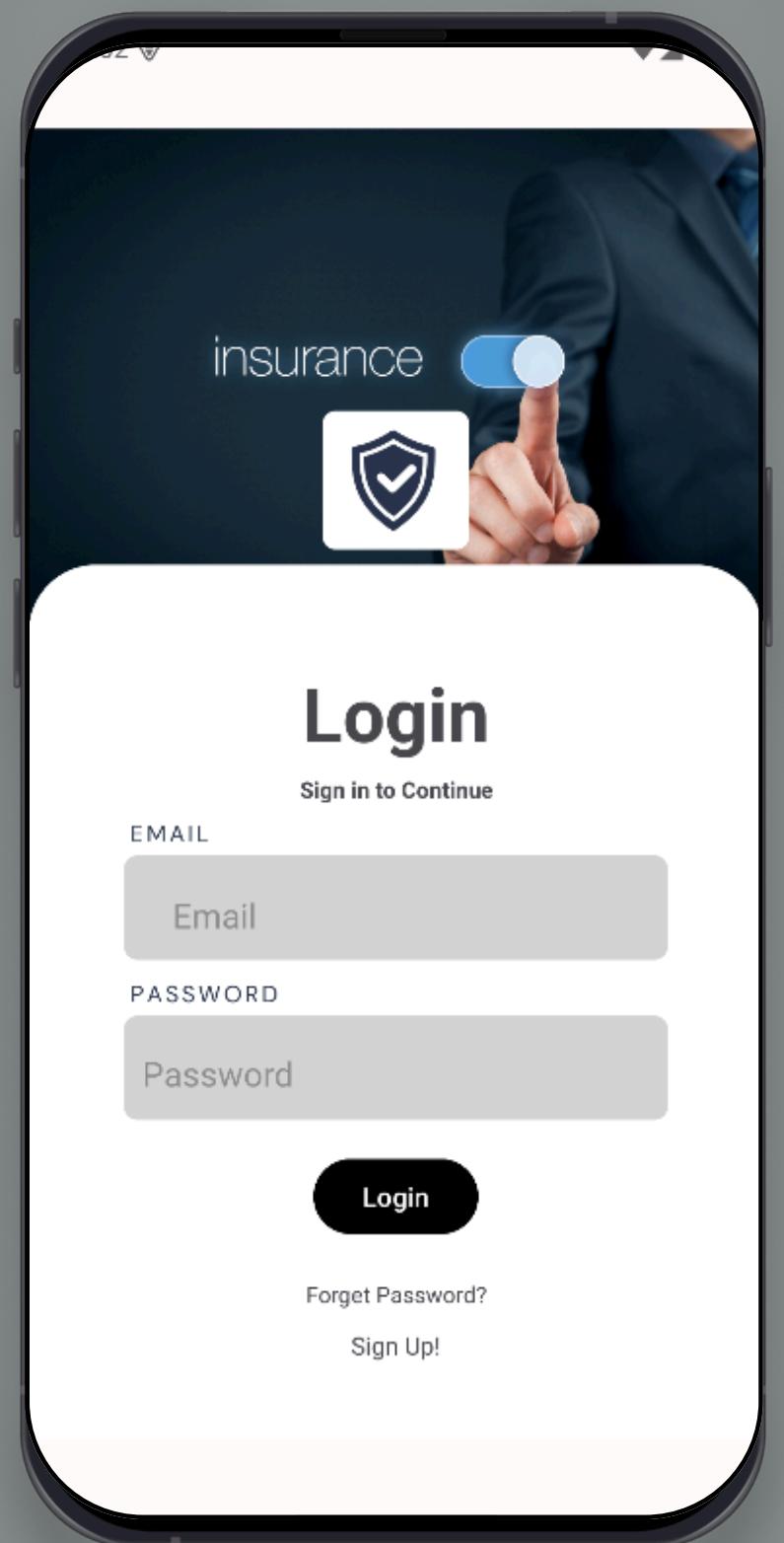
Here we add the new vehicle details of the user.

● CardView

This card view shows the latest vehicle details submitted by the user.

● About Policies

Here the list of the policies mentioned. That are selected by users.



PRODUCT DEMO

Login Screen

michaeljoshua.p1504@gmail.com

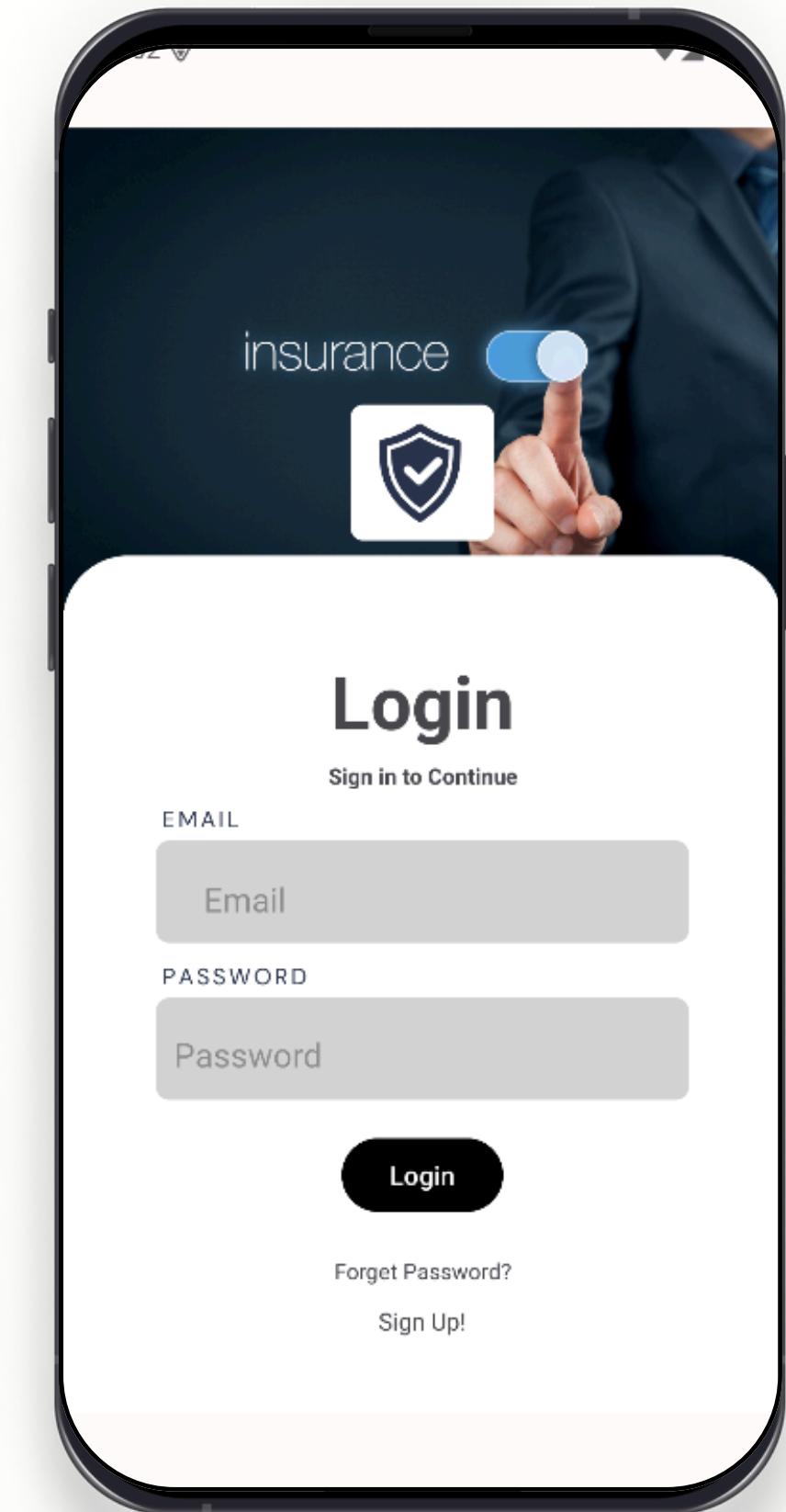
Vehicle Insurance App

VEHICLE INSURANCE APP

Login Screen

PURPOSE OF MAINACTIVITY

MainActivity.java serves as the entry point of the app, handling user login authentication and navigation to the home screen upon successful login. It incorporates methods for validating user credentials, making API calls to authenticate users against the backend server, and securely managing user sessions using SharedPreferences.



App Logo

Login Title

Email

user Enters

Password

User Enters

Login Button

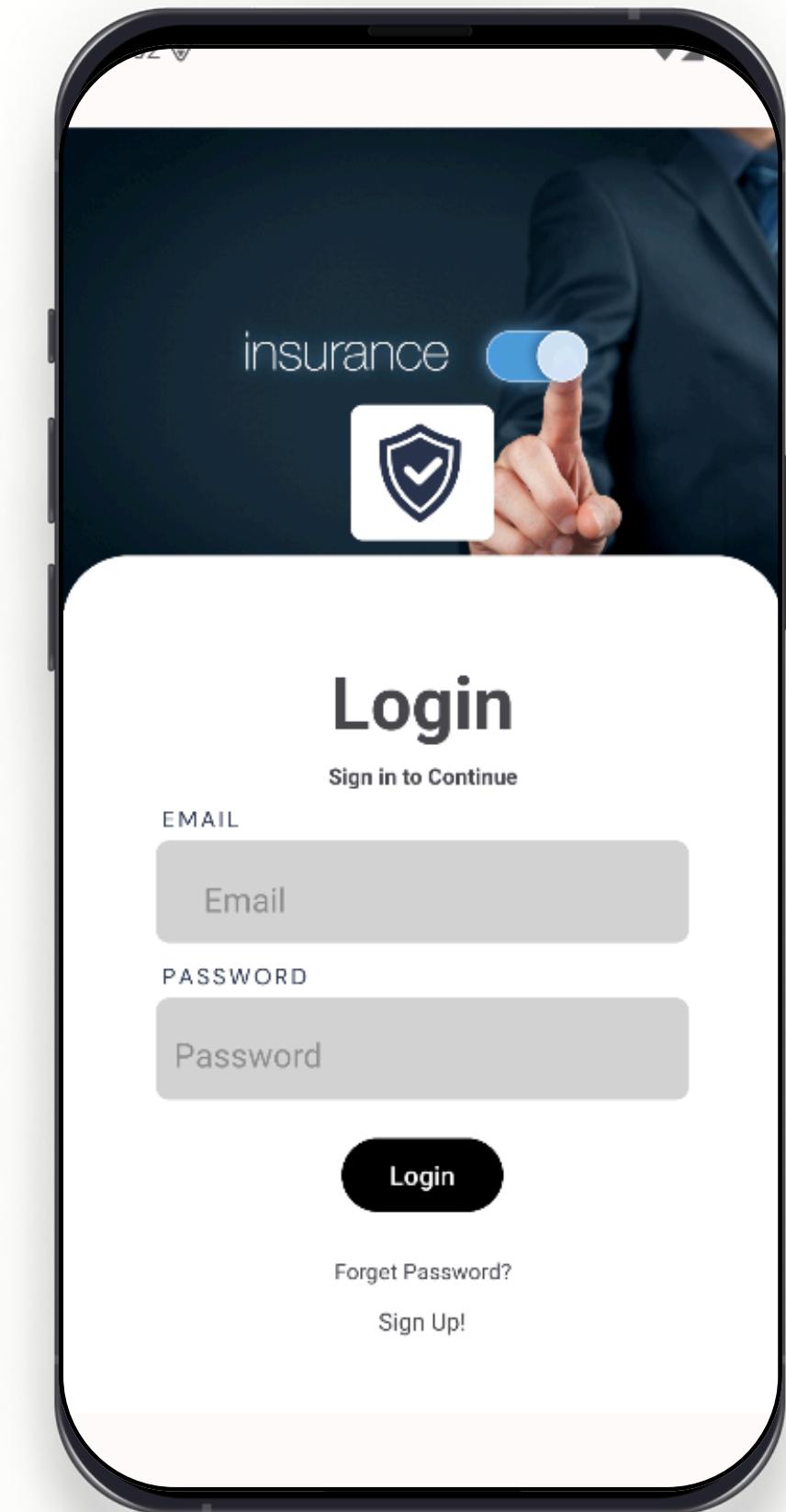
Signup Button

VEHICLE INSURANCE APP

Login Screen

LAYOUT OF THE MAIN ACTIVITY

activity_main.xml defines the layout for the login screen, featuring EditText fields for user input, a Button for login submission, and TextViews for error messages. It utilizes RelativeLayout as the root layout to manage UI element positioning dynamically. The XML file also includes attributes for styling and aligning UI components effectively.



App Logo

Login Title

Email

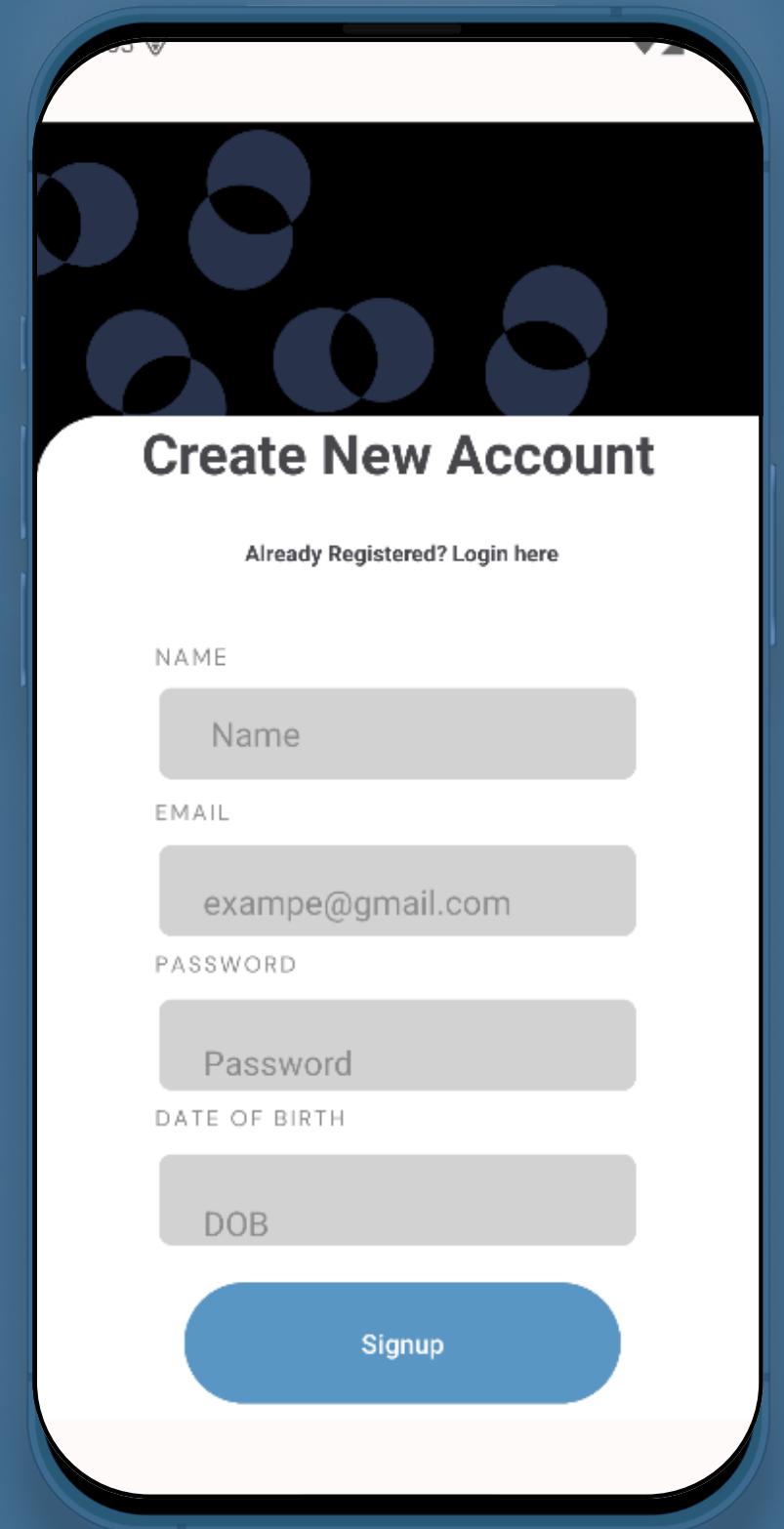
user Enters

Password

User Enters

Login Button

Signup Button



PRODUCT DEMO

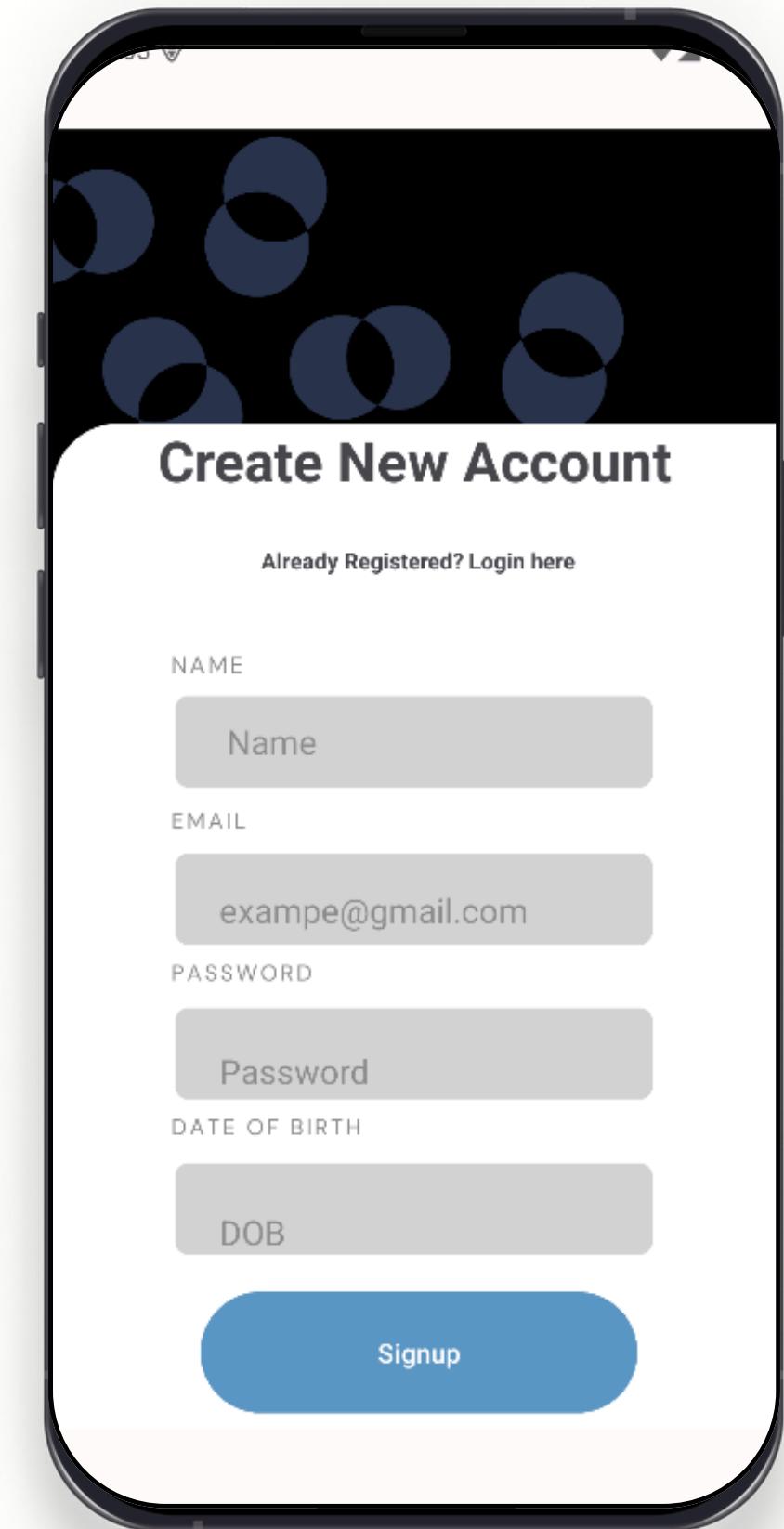
Signup Screen

VEHICLE INSURANCE APP

Signup Screen

PURPOSE OF SIGNUPACTIVITY

SignUpActivity.java facilitates user registration by capturing user details such as name, email, password, and date of birth. It validates user inputs, communicates with the backend server to create new user accounts, and navigates users to the home screen upon successful registration. Error handling mechanisms ensure a smooth signup process.



Title

Navigation to login

Name

Email

Password

DOB

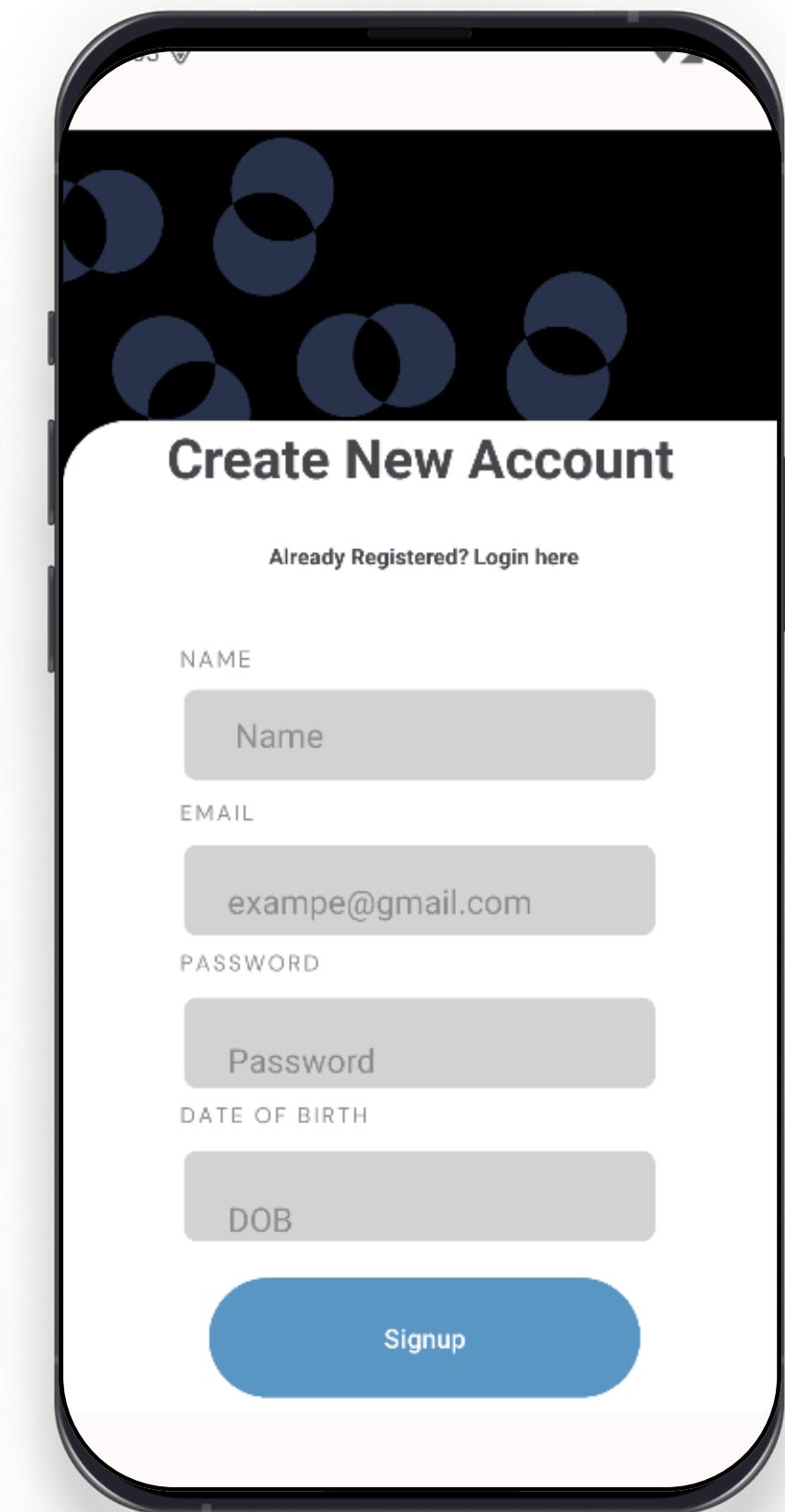
Signup Button

VEHICLE INSURANCE APP

Signup Screen

LAYOUT OF THE SIGNUP ACTIVITY

activity_sign_up.xml presents a registration form comprising EditText fields for user inputs, a Button for submitting registration details, and TextViews for displaying validation errors. The layout is designed using ConstraintLayout to optimize screen utilization and maintain a responsive UI across different Android devices.



Title

Navigation to login

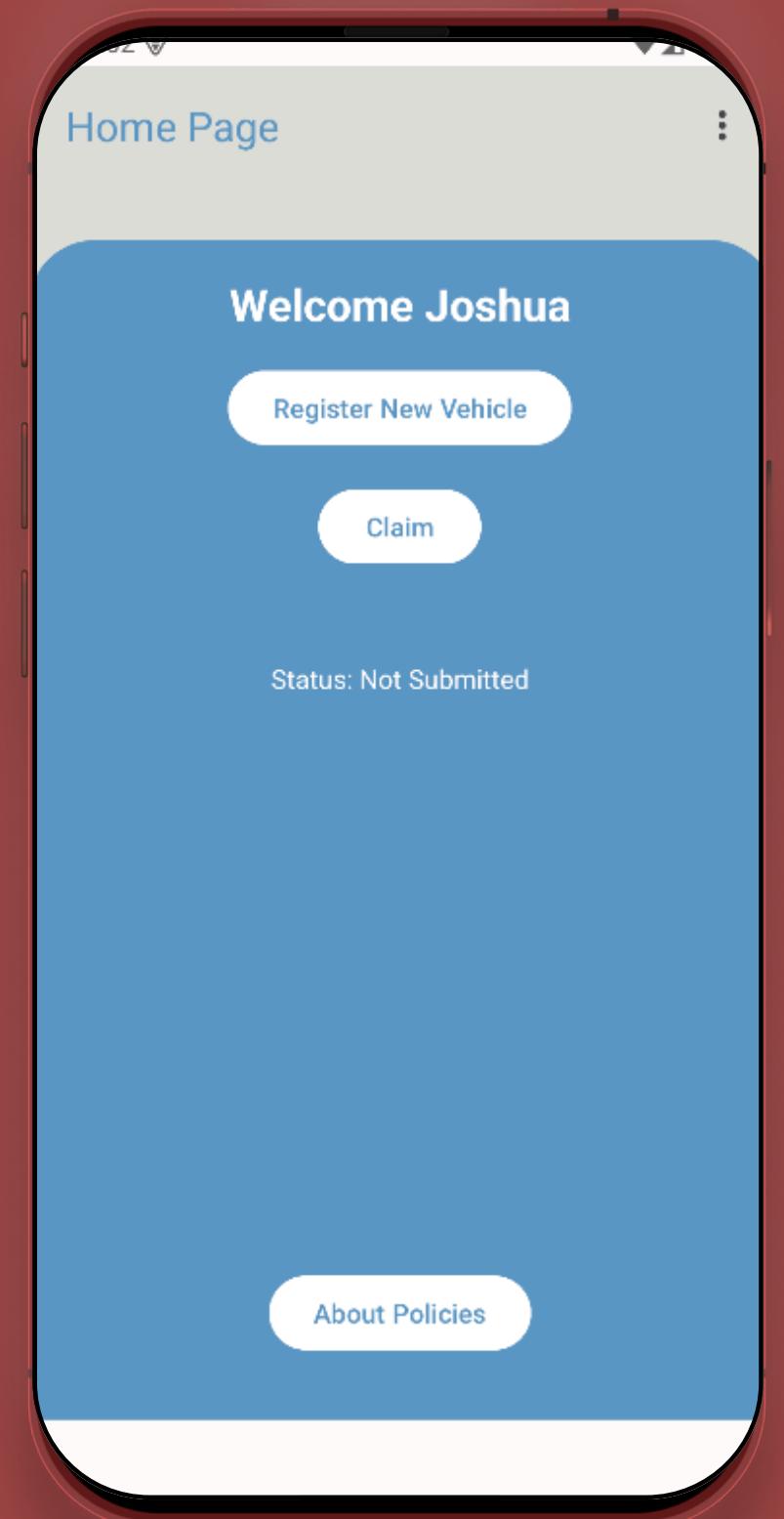
Name

Email

Password

DOB

Signup Button



PRODUCT DEMO

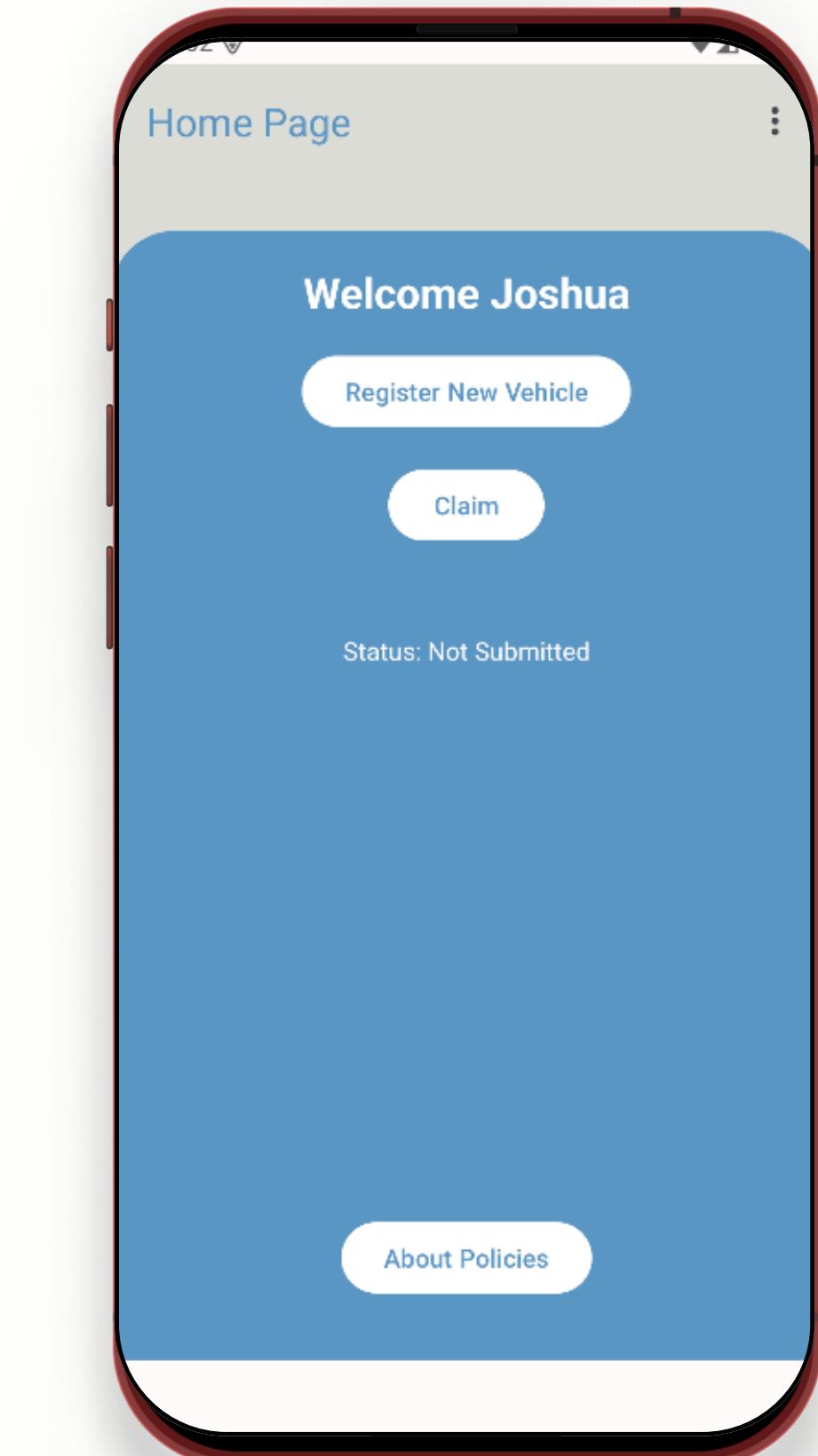
Home Page Screen

VEHICLE INSURANCE APP

HomePage Screen

PURPOSE OF HOMEPAGE ACTIVITY

HomePageActivity.java serves as the central hub of the app, displaying personalized information such as user name, registered vehicle details, and insurance policy status. It integrates with backend APIs to fetch and display dynamic data, supports navigation to other app features like vehicle registration and claim submission, and provides a seamless user experience.



Menu

Welcome
Username

CardView
Displays Vehicle
details

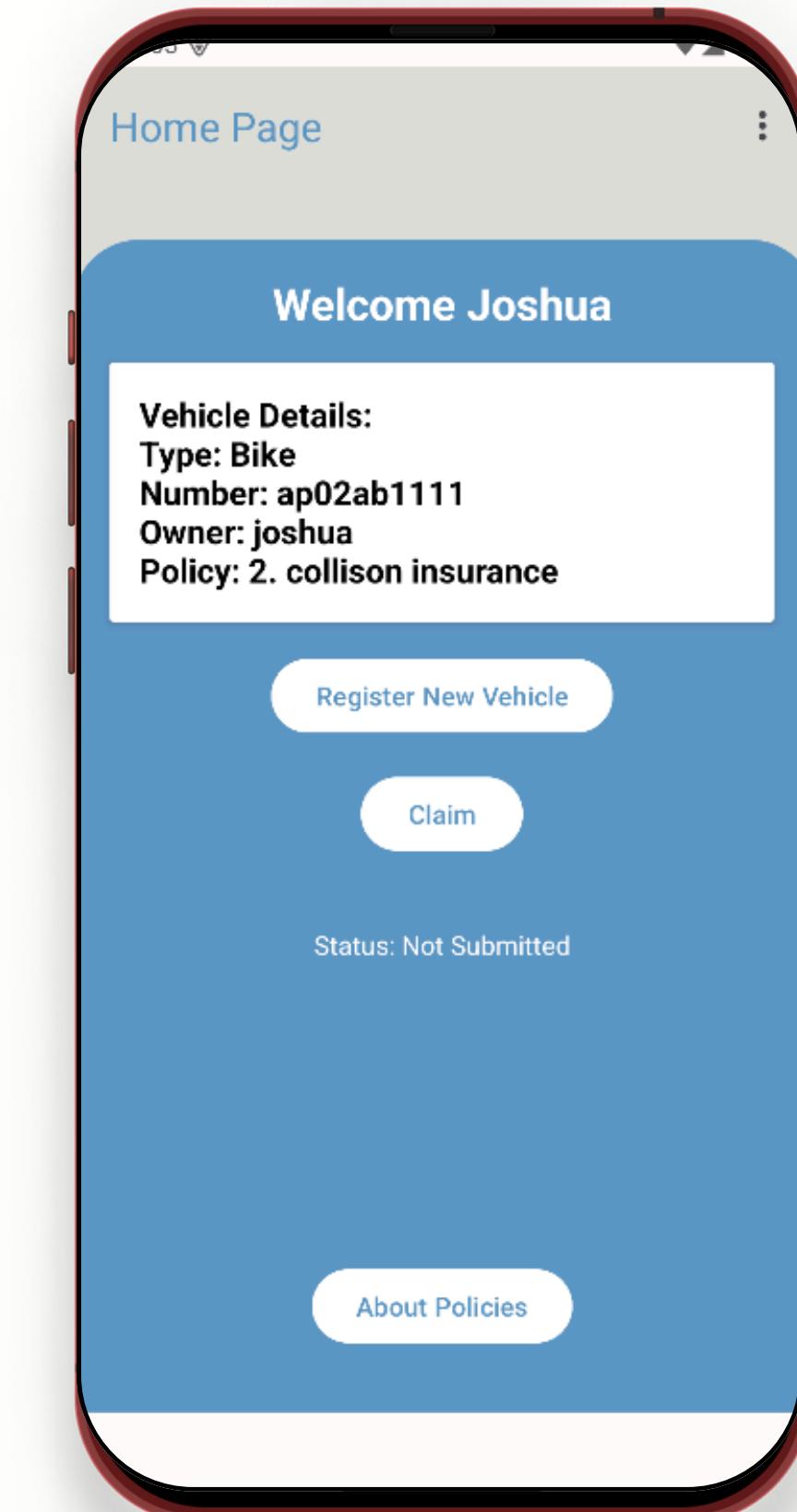
Register New
Vehicle
About Policies

VEHICLE INSURANCE APP

HomePage Screen

LAYOUT OF THE HOME PAGE ACTIVITY

activity_home_page.xml defines the visual layout for HomePageActivity, featuring a Toolbar for navigation and branding, TextViews for displaying user-specific information, CardViews for presenting registered vehicle details, and Buttons for initiating new actions such as vehicle registration and claim submission. The layout is optimized for usability and clarity, ensuring a user-friendly interface.



Menu

Welcome

Username

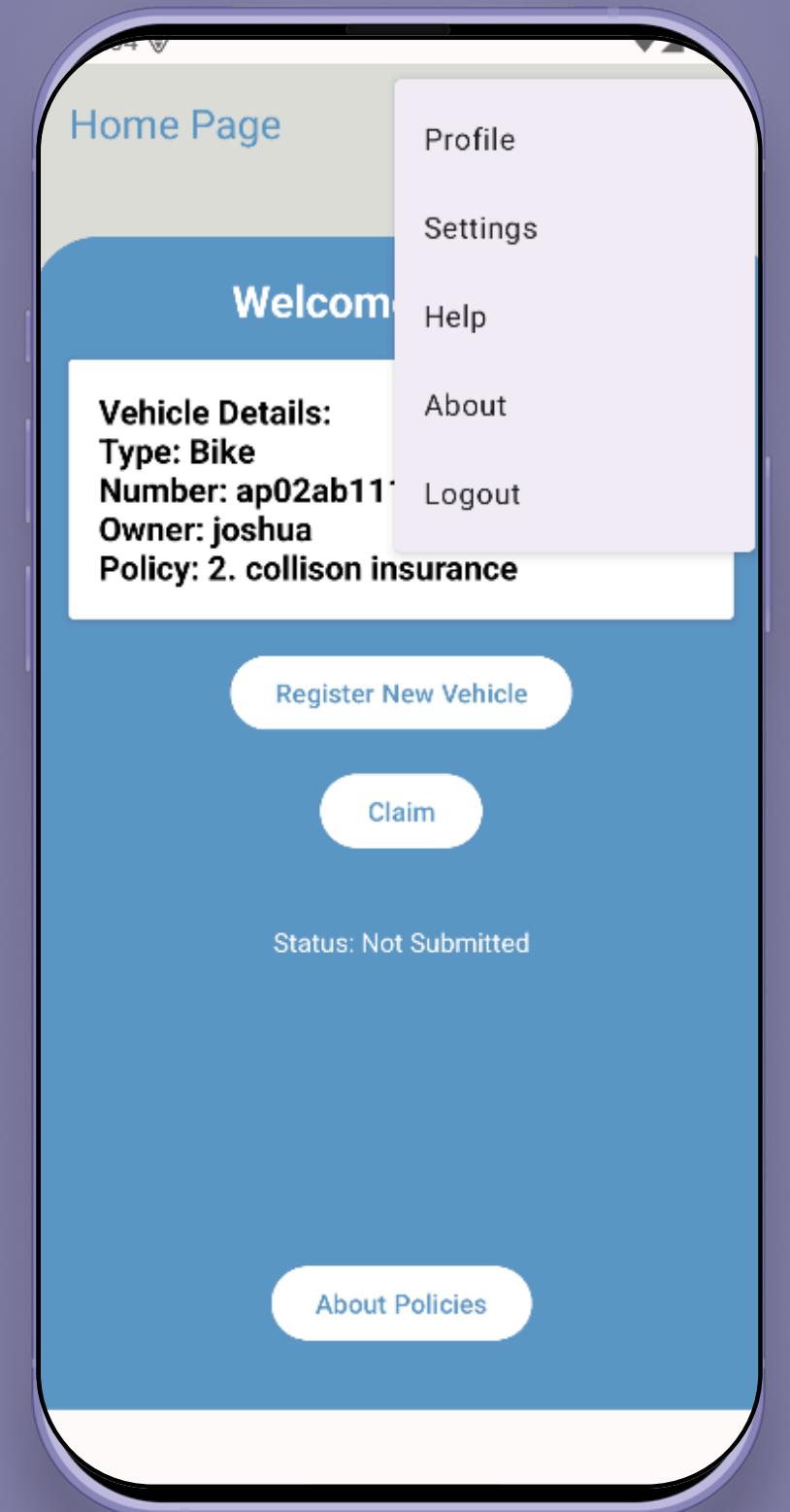
CardView

Displays Vehicle
details

Register New

Vehicle

About Policies



PRODUCT DEMO

Menu

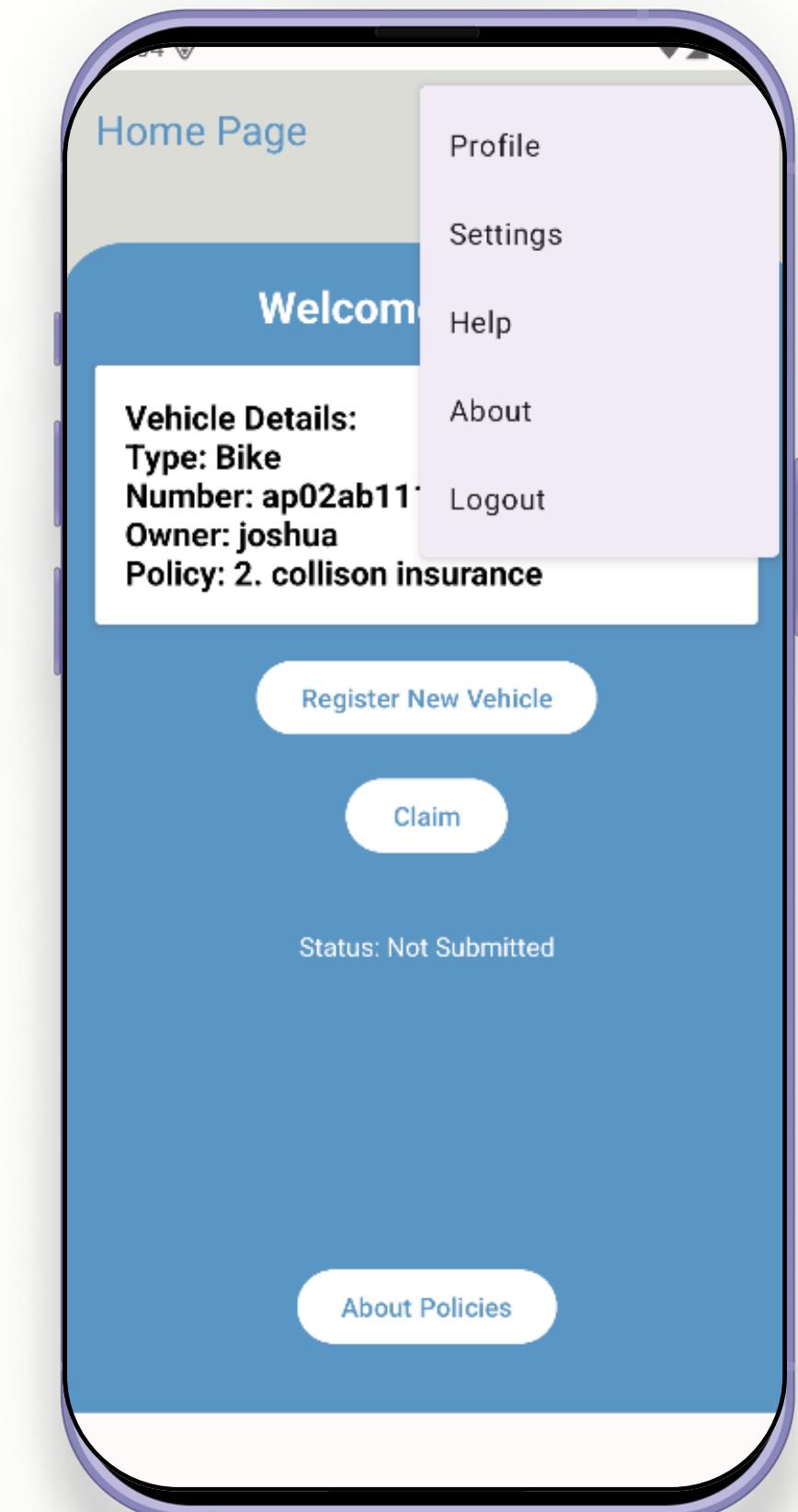
VEHICLE INSURANCE APP

Menu

- Purpose: Enhance user experience by providing easy access to key functionalities and information.
- Components: Profile, Settings, Help, About, Logout.
- Placement: Located in the toolbar or navigation drawer for quick access.

IMPLEMENTATION

- Technical Aspects: Integration using Android Studio and XML layouts.
- Navigation: Implementing click listeners and intents for menu items.
- Testing: Verify functionality across different devices and scenarios.



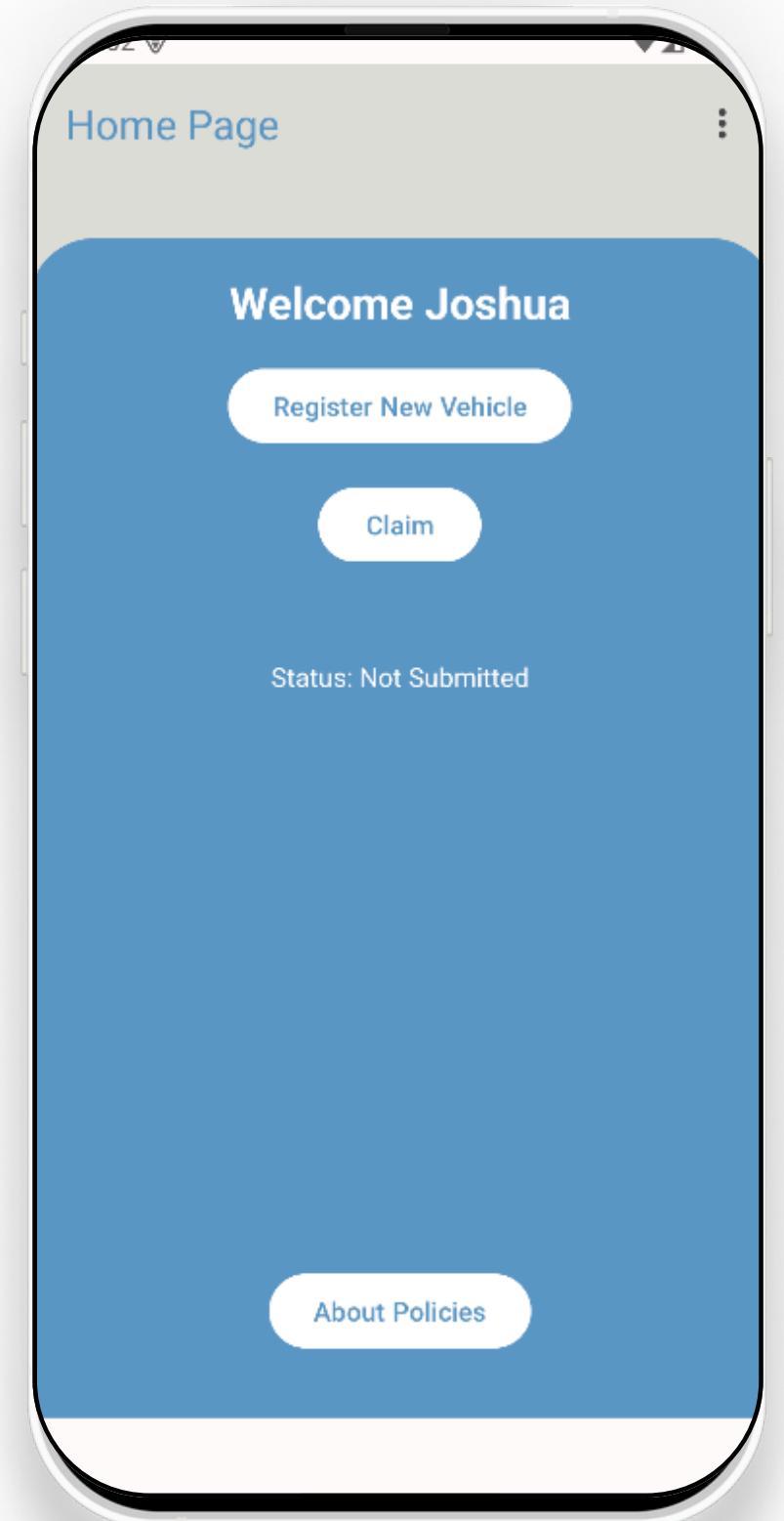
Profile

Settings

Help

About

Logout



Welcome Name

WHEN THE USER LOGINS THE NAME WHICH IS GIVEN IN THE SIGNUP WILL BE DISPLAYED IN THE HOMEPAGE WITH THE WELCOME MESSAGE.

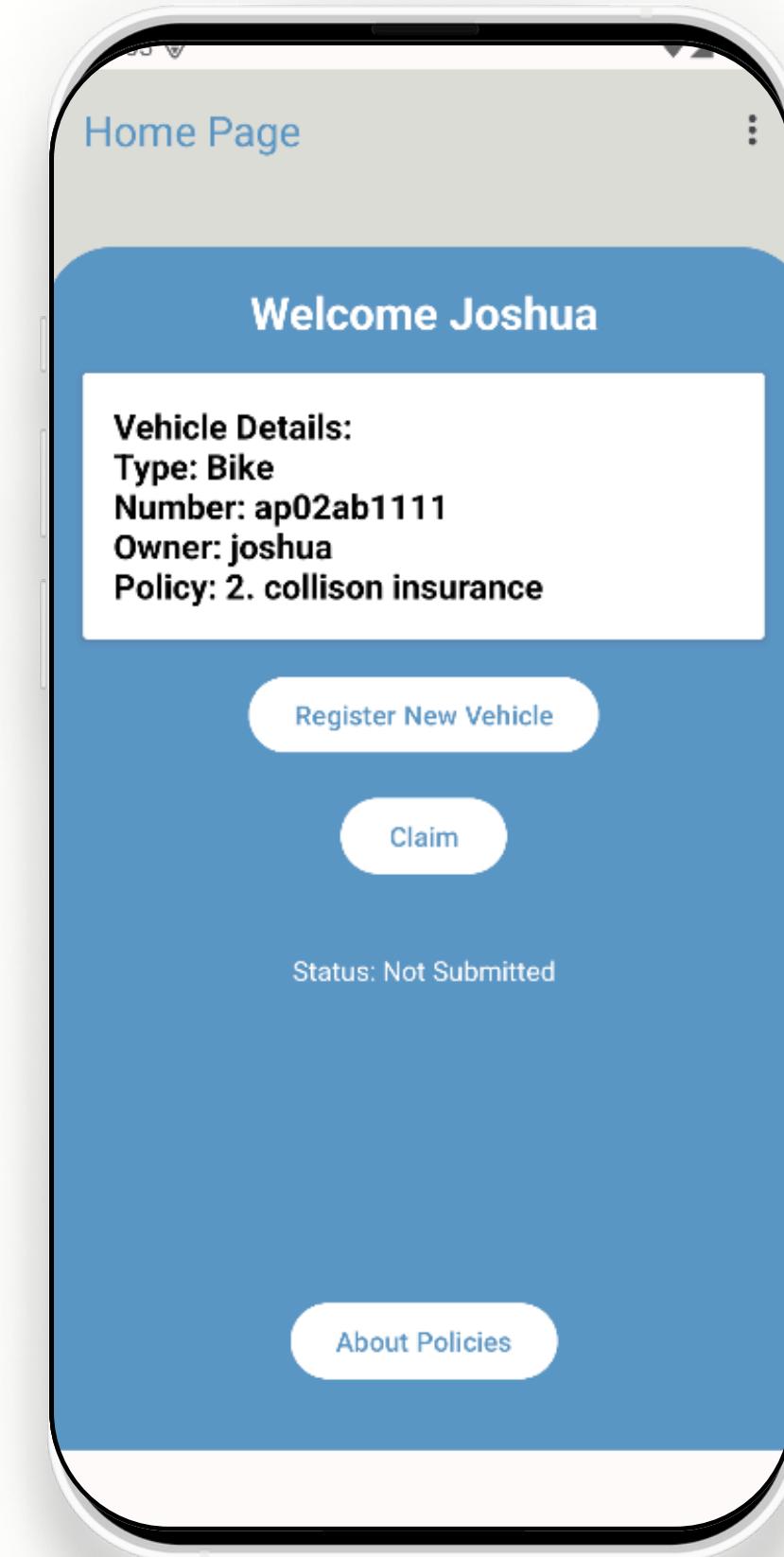
CardView

INTRODUCTION TO CARDVIEW

- Purpose: Enhance visual appeal and organization of information on the homepage.
- Functionality: Display essential data in a structured format to ease the user about the details.
- Benefits: Provides a consistent and modular layout for displaying different types of information.

IMPLEMENTATION DETAILS

- XML Layout: Defines CardView attributes such as width, height, margins.
- Content: Includes text fields for vehicle details like type, number, owner name and Policy type.
- Dynamic Data Binding: Populate CardViews dynamically based on user inputs or database queries.



CardView
Displays
Vehicles
Details
upon Login

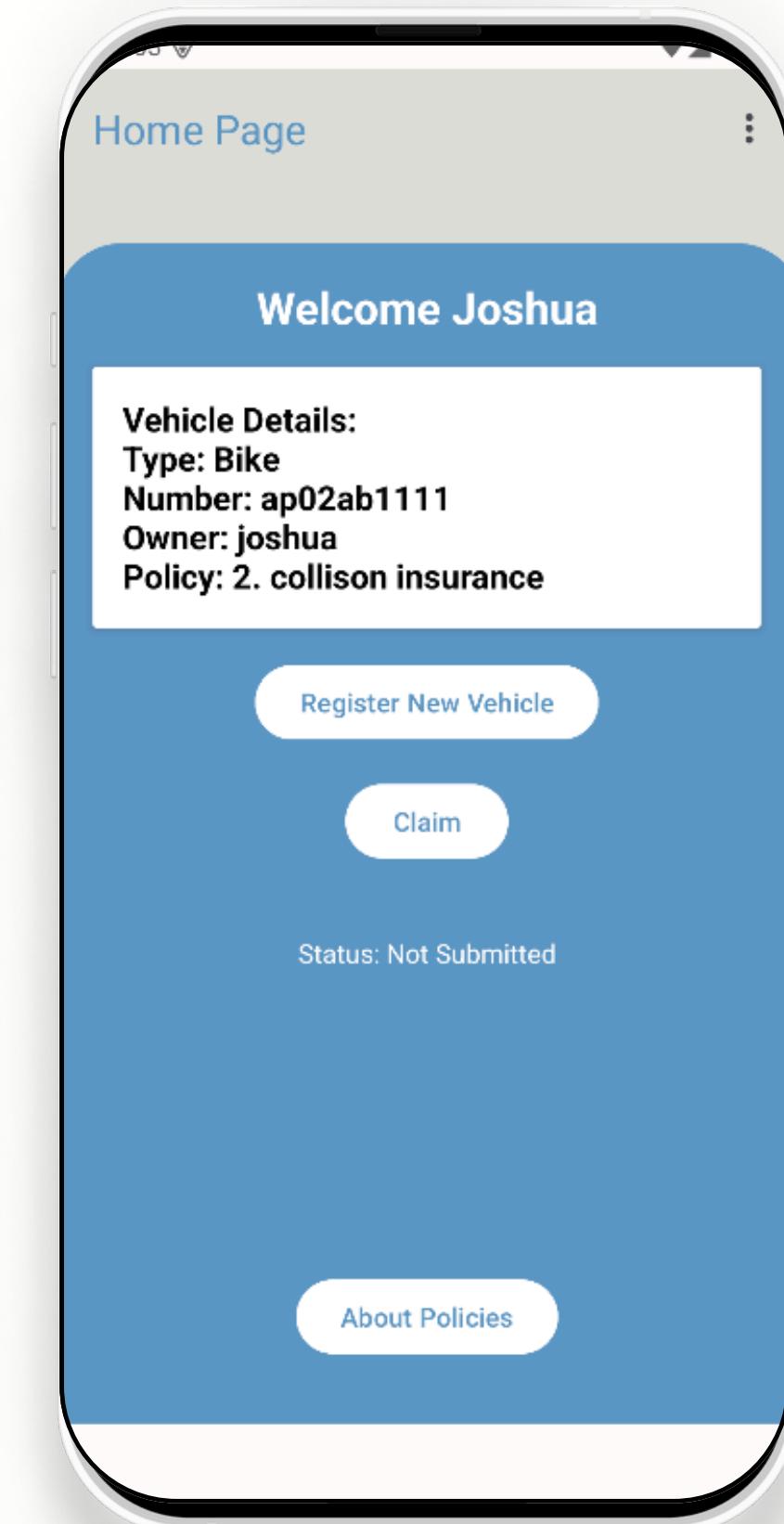
Vehicle Details

PURPOSE OF REGISTER VEHICLE ACTIVITY

RegisterVehicleActivity.java facilitates the registration of new vehicles into the app's database. It captures vehicle details such as type, number, and owner name through EditText fields, validates user inputs, and communicates with backend services to store vehicle information securely. The activity enhances user convenience by streamlining the vehicle registration process.

PURPOSE OF REGISTER VEHICLE ACTIVITY

RegisterVehicleActivity.java facilitates the registration of new vehicles into the app's database. It captures vehicle details such as type, number, and owner name through EditText fields, validates user inputs, and communicates with backend services to store vehicle information securely. The activity enhances user convenience by streamlining the vehicle registration process.



**Register New
Vehicle Button
Navigates to
the new page**

VEHICLE INSURANCE APP

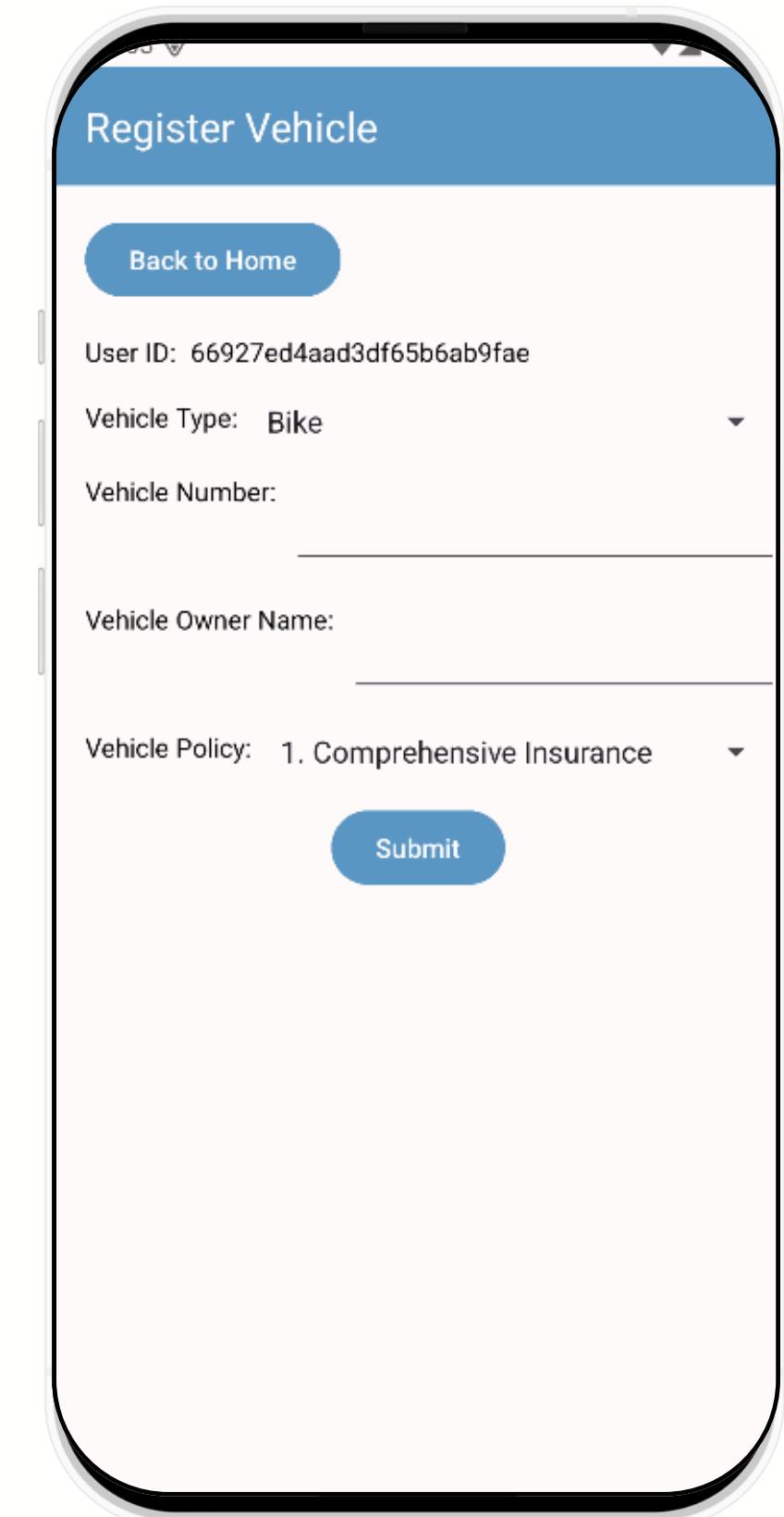
Register Vehicle

This screen provides the fields to register a new vehicle.

User ID is unique which is created on the serverside to specify the particular. This ID helps to fetch the any details of the user required.

Vehicle Type will be entered by the user. vehicle number and also the type of policy.

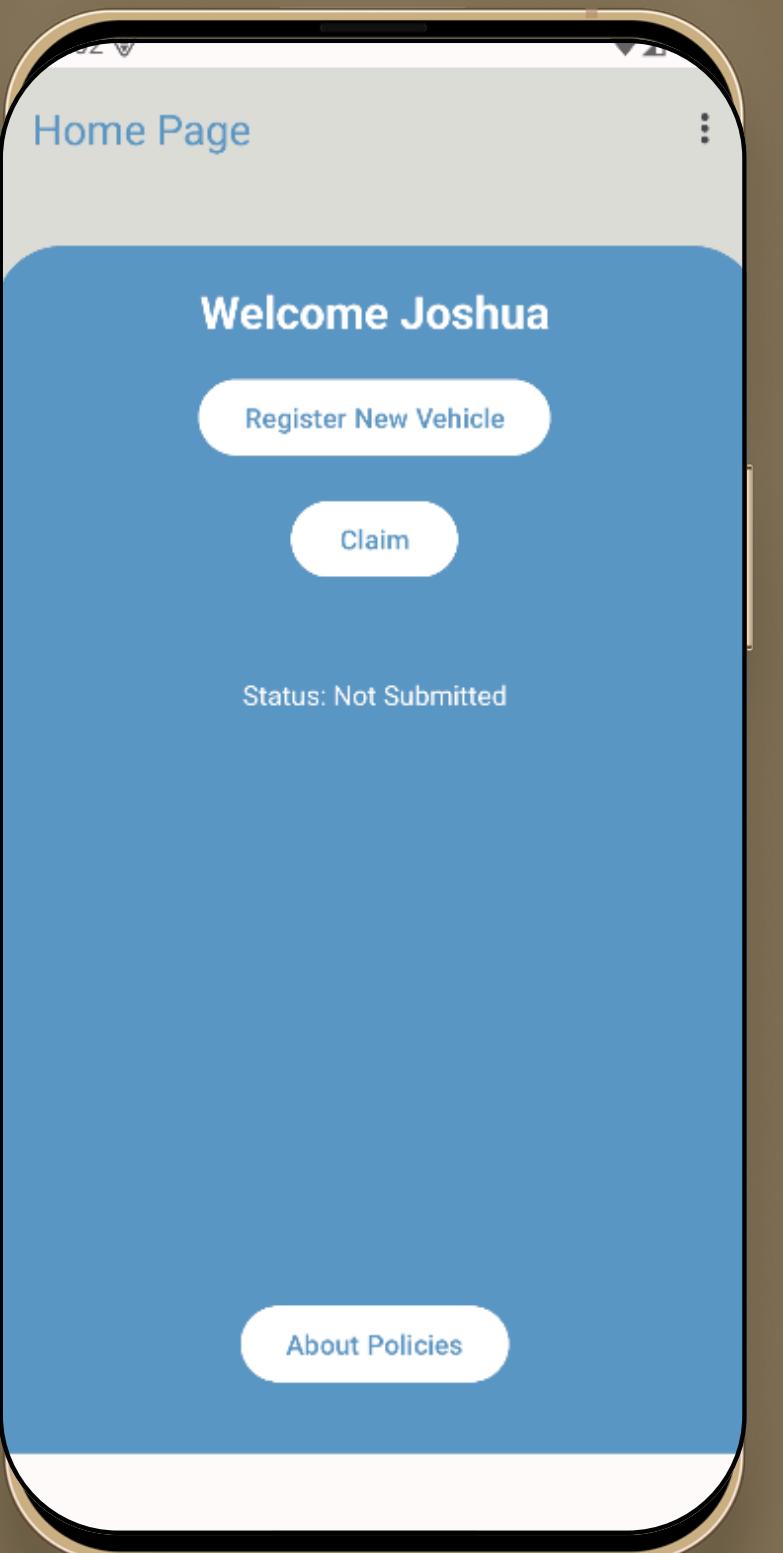
All details will be saved in the session manager and will be displayed back to the homepage.



New Page

**Enters vehicles
Details**

Submit



PRODUCT DEMO

Policy Details

Policy Details

PURPOSE OF POLICY ACTIVITY

PolicyActivity.java displays insurance policy details within the app, providing users with comprehensive information regarding coverage, terms, and conditions. It retrieves policy content from the backend server, dynamically populates a TextView with policy text, and includes navigation functionality back to the home screen for user convenience.

LAYOUT OF THE POLICY ACTIVITY

activity_policy.xml defines the layout for PolicyActivity, featuring a Toolbar for navigation, a ScrollView for displaying policy text, and a Button for returning to the home screen. The XML layout prioritizes readability and accessibility, ensuring that policy details are presented clearly to users and facilitating seamless navigation between app functionalities.



Back Button

Tool Bar

Policy Details

VEHICLE INSURANCE

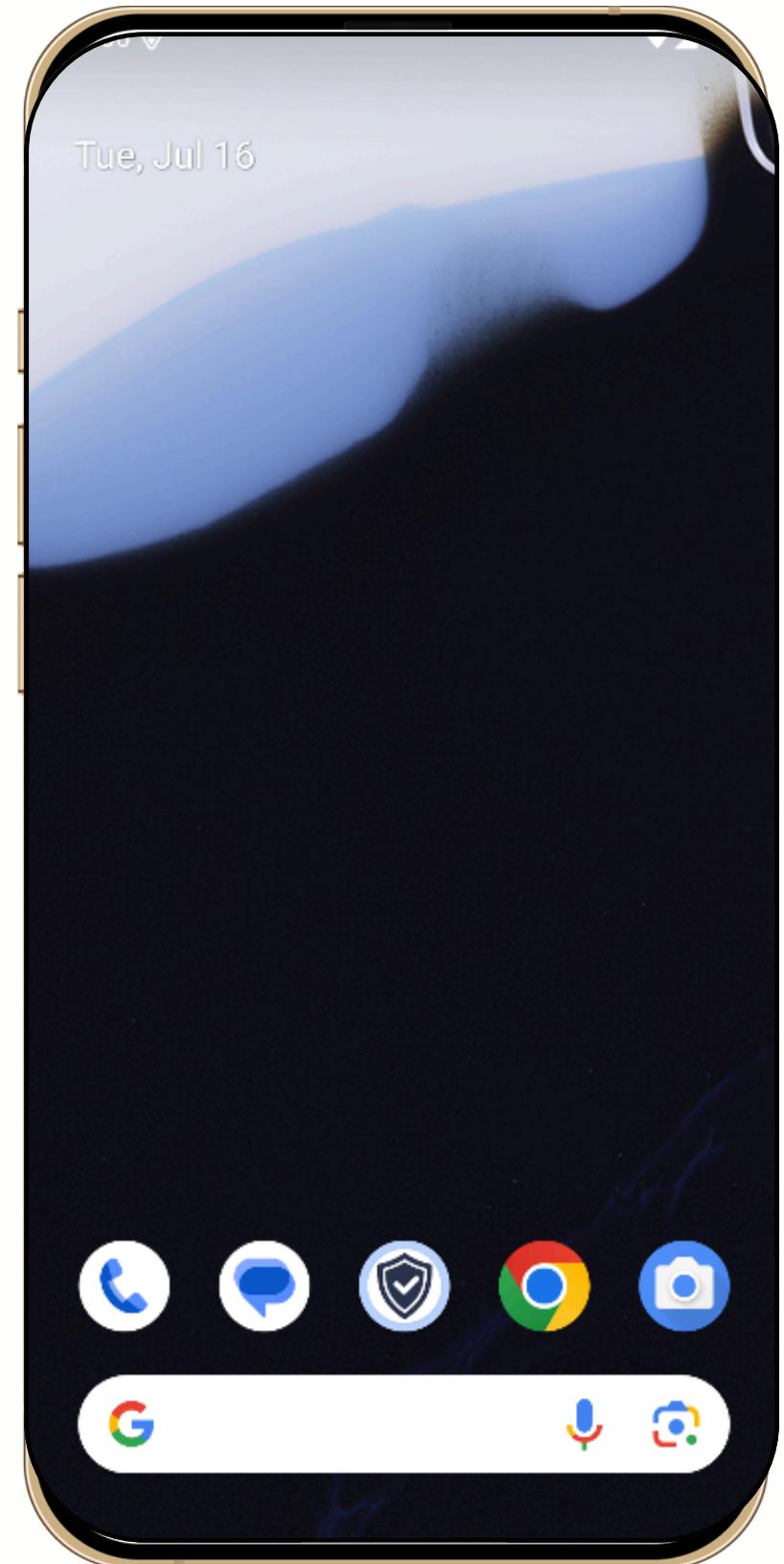
Splash Screen

PURPOSE OF SPLASHACTIVITY

SplashActivity.java provides a welcoming introduction to the app, displaying a logo or animation briefly upon app launch. It initializes essential app components in the background, such as checking user authentication status or loading necessary resources, before seamlessly transitioning to the main activity. This enhances user engagement and sets the stage for a smooth user experience.

LAYOUT OF THE SPLASH SCREEN ACTIVITY

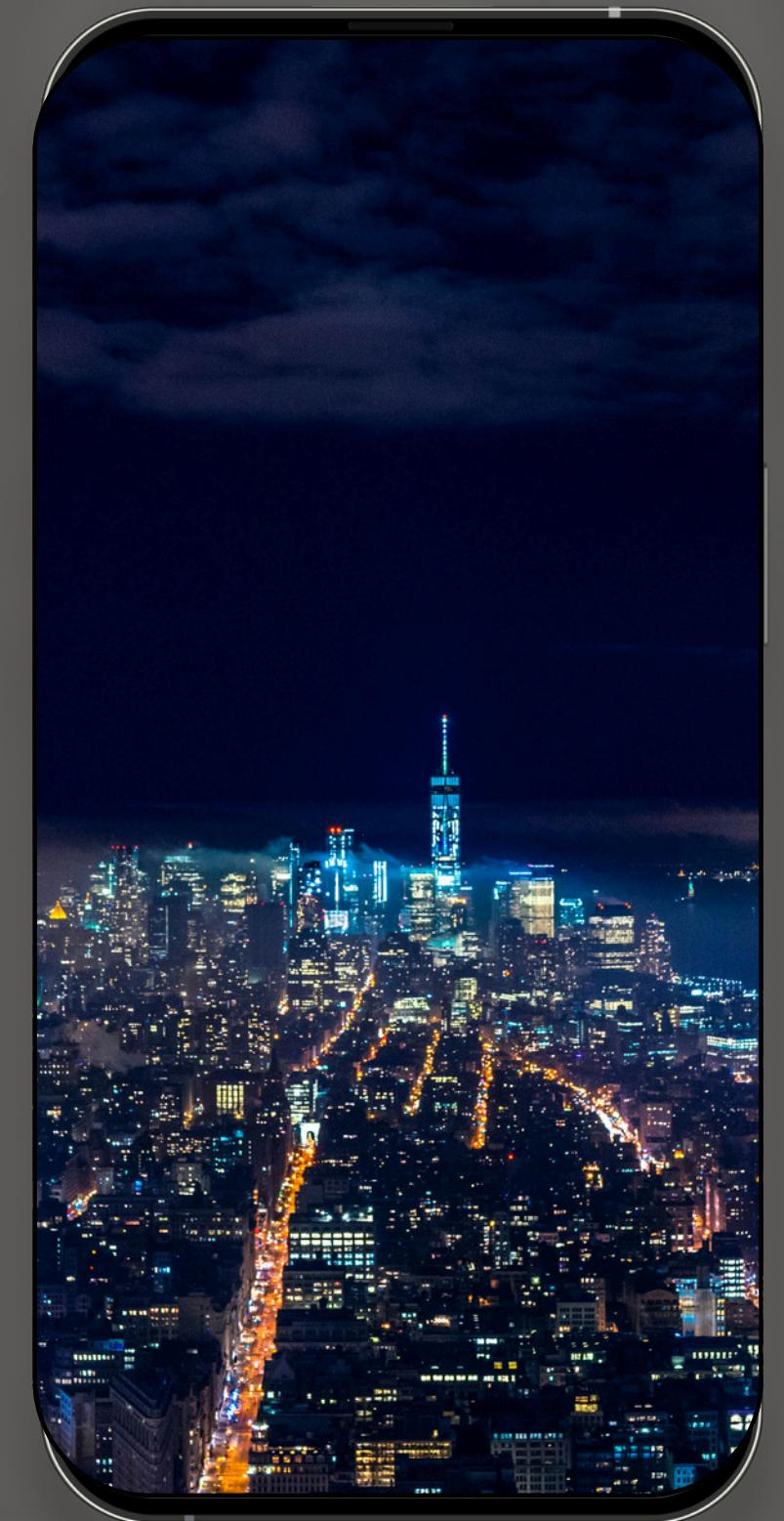
activity_splash.xml defines the layout for SplashActivity, typically featuring an ImageView or VideoView to display a logo or animation. It employs ConstraintLayout to position and scale visual elements optimally, ensuring compatibility across various Android devices and screen sizes. The layout prioritizes aesthetics and brand identity, making a memorable first impression on users.



HomeScreen

App is opened

Splash screen displays



PRODUCT DEMO

Backend Integration

michaeljoshua.p1504@gmail.com

Vehicle Insurance App

VEHICLE INSURANCE APP

Backend Integration

Backend integration involves using Node.js with Express.js to create RESTful APIs that handle data transactions between the mobile app and MongoDB database. APIs manage user authentication, vehicle registration, claim submission, and policy retrieval, ensuring seamless communication and data synchronization between frontend and backend components.

- **user.js**

There are the main files for the backend. The files are deployed in the render.com and got the url and connected with the front-end.

- **vehicle.js**

- **claim.js**

- **server.js**

Backend Code - 1

EXPLANATION OF USER SCHEMA AND MODEL

user.js defines the schema and model for managing user data within the MongoDB database. It specifies fields such as name, email, password, and date of birth, enforcing data integrity and security measures through schema validation and encryption techniques. The user model supports CRUD operations for user management, facilitating robust backend functionality.

EXPLANATION OF VEHICLE SCHEMA AND MODEL

vehicle.js outlines the schema and model structure for storing vehicle information in MongoDB. It includes fields like vehicle type, number, owner name, and policy details, enabling efficient storage and retrieval of vehicle data through API endpoints. The vehicle model supports operations for adding, updating, and querying vehicle records, enhancing app functionality related to vehicle management.

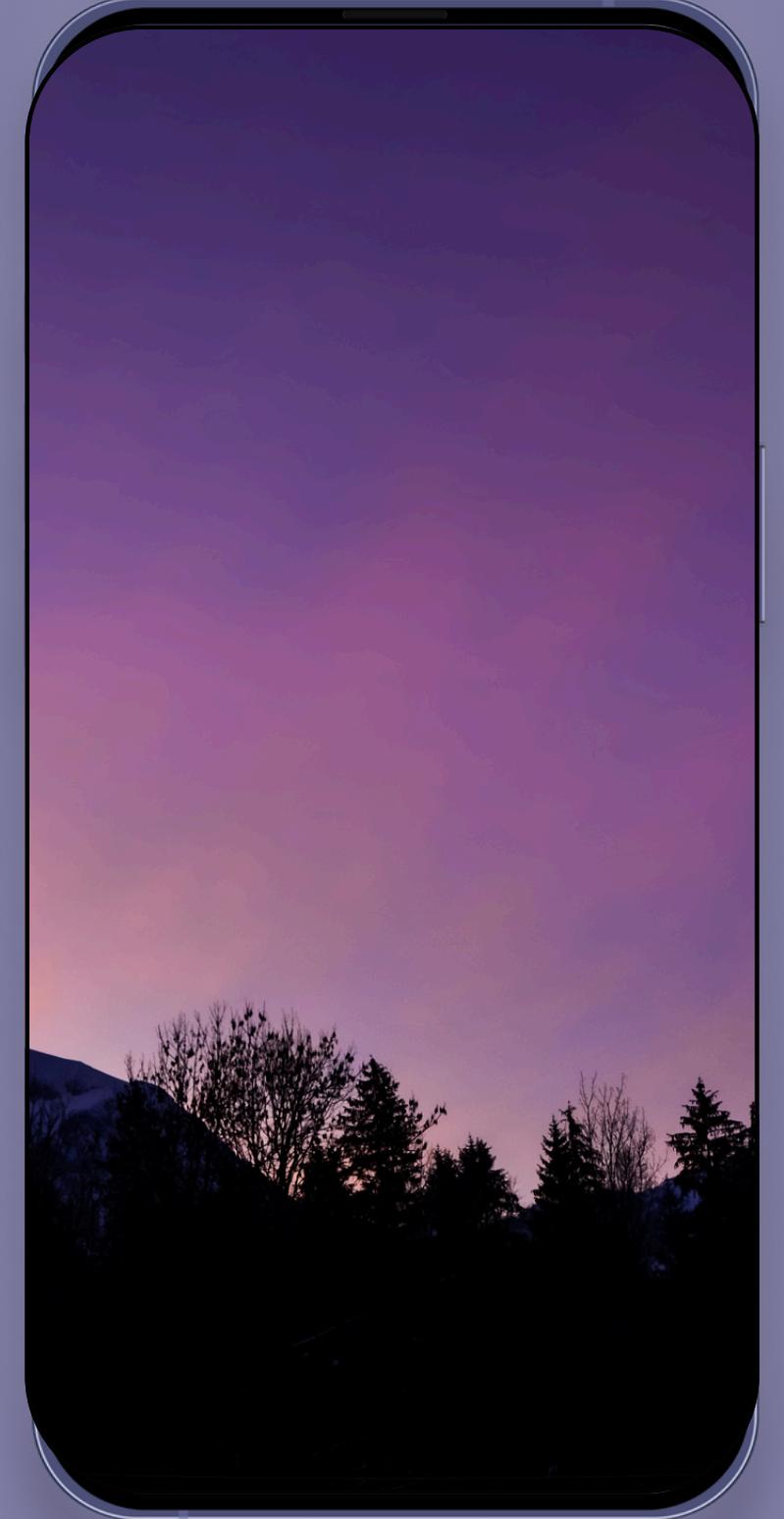
Backend Code - 2

EXPLANATION OF CLAIM SCHEMA AND MODEL

claim.js defines the schema and model for managing insurance claim submissions within the MongoDB database. It incorporates fields such as claim date, description, status, and associated vehicle details, facilitating streamlined processing and tracking of insurance claims through RESTful APIs. The claim model supports CRUD operations for handling claim submissions and updates, ensuring data accuracy and accessibility.

OVERVIEW OF SERVER SETUP AND API ENDPOINTS

server.js configures the backend server using Node.js and Express.js, defining API routes for handling requests from the mobile app. It establishes endpoints for user authentication, vehicle registration, claim submission, and policy retrieval, implementing middleware for request validation, error handling, and secure data transmission. The server architecture supports scalability and performance optimization, ensuring reliable operation of the Vehicle Insurance App.



PRODUCT DEMO

API Integration

michaeljoshua.p1504@gmail.com

Vehicle Insurance App

VEHICLE INSURANCE APP

API Integration

EXPLANATION OF APICLIENT AND APISERVICE

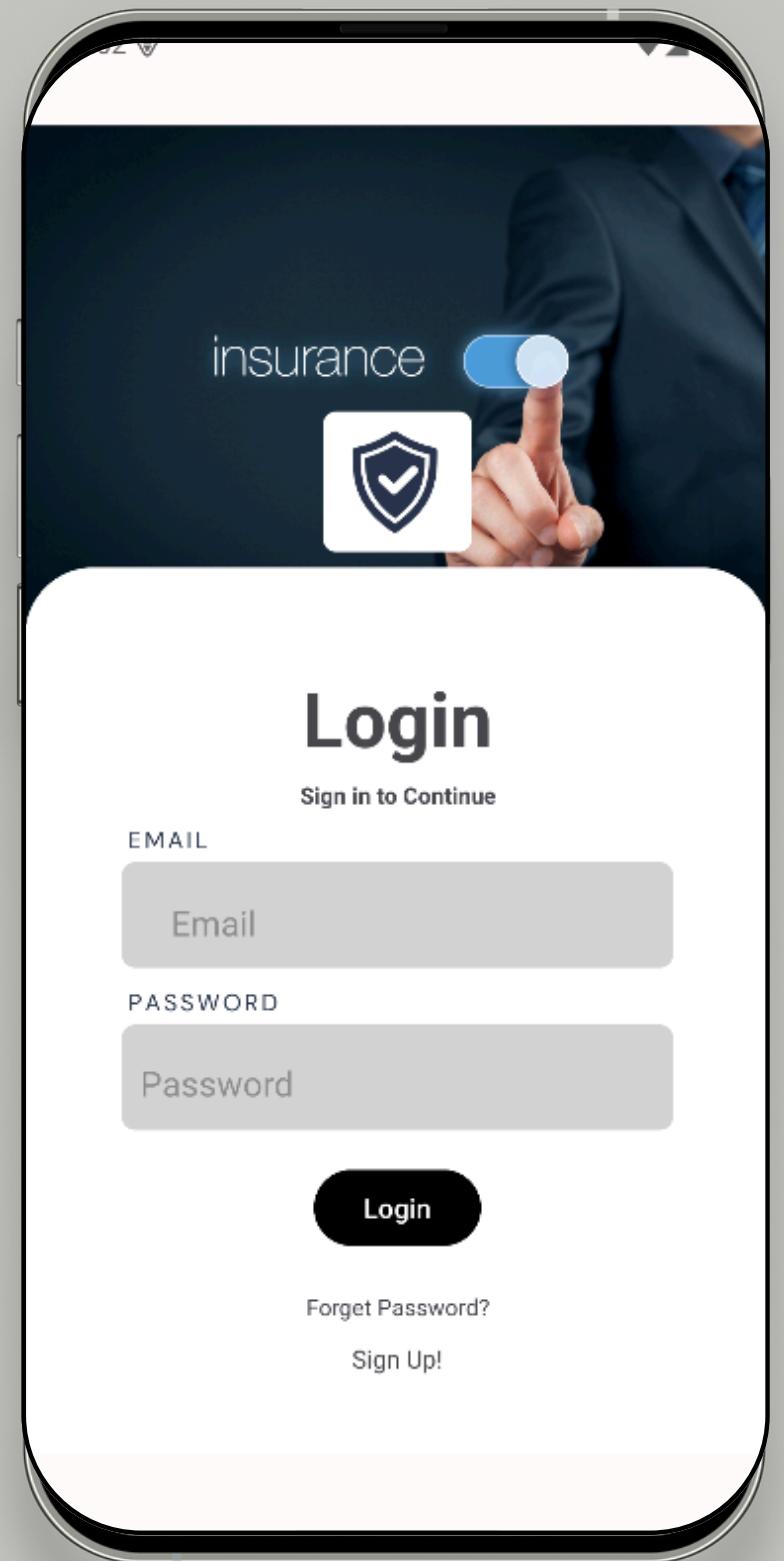
- APIClient and ApiService facilitate seamless integration between the Android app and backend server through RESTful API calls. APIClient manages network requests, including authentication tokens and error handling, while ApiService defines methods for interacting with specific API endpoints related to user management, vehicle registration, claim submission, and policy retrieval. This modular approach enhances code reusability and simplifies maintenance, promoting efficient data exchange and synchronization between frontend and backend components.

VEHICLE INSURANCE APP

SharedPreferences

OVERVIEW OF SHARED PREFERENCES USAGE

SharedPreferences in Android stores key-value pairs locally on the device, facilitating persistent data storage for app preferences, user sessions, and configuration settings. In the Vehicle Insurance App, SharedPreferences manage user authentication tokens, session information, and UI state persistence, ensuring a seamless user experience across app sessions and device restarts. This lightweight data storage solution optimizes app performance and enhances user convenience by retaining critical app data securely.

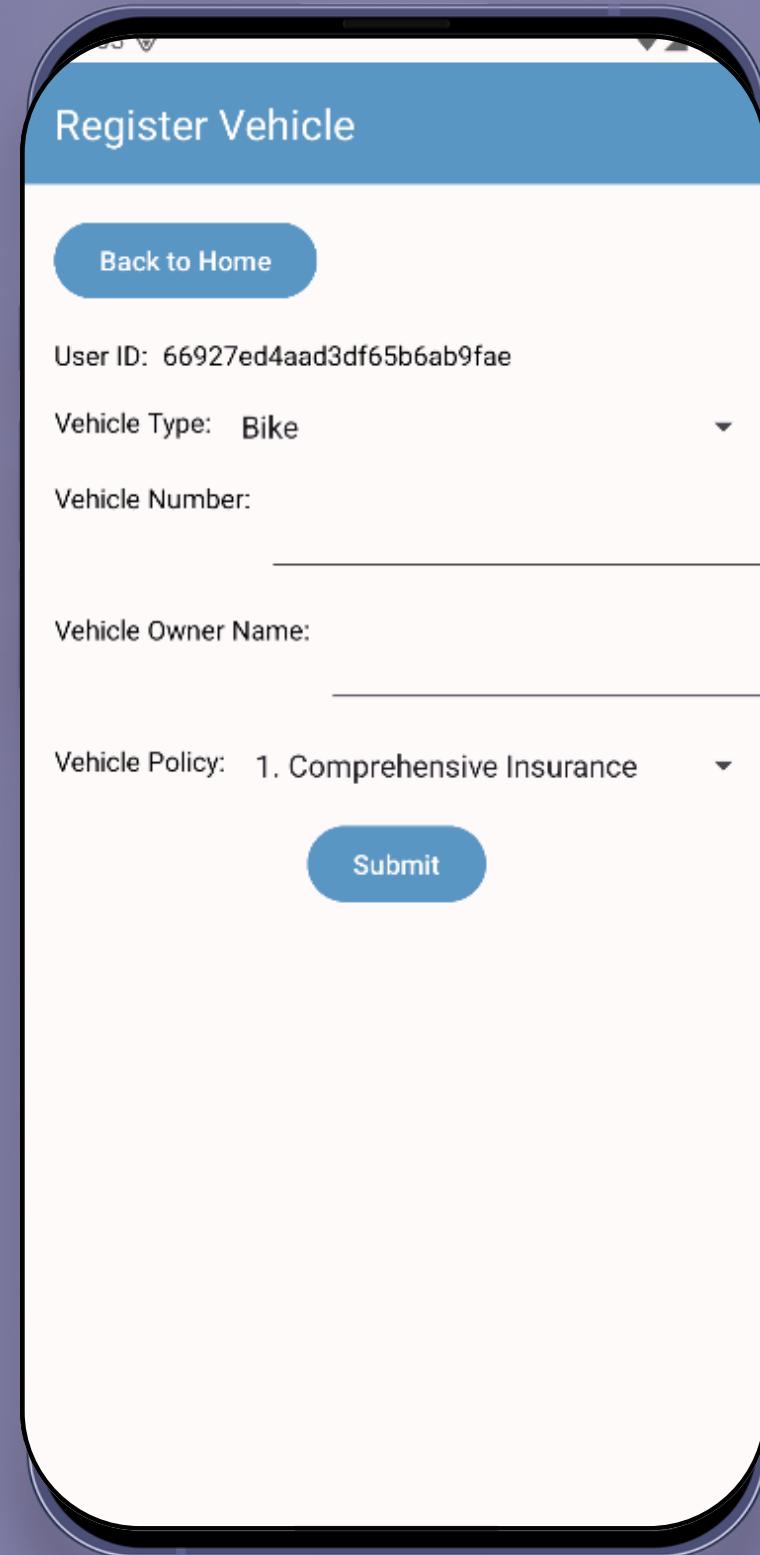


PRODUCT DEMO

Login and Signup Flow

EXPLANATION OF USER AUTHENTICATION PROCESS

The login and signup flow in the Vehicle Insurance App ensures secure user authentication and registration processes. Upon launching the app, users input their credentials via `MainActivity`, where data validation and authentication requests are processed. `SignUpActivity` enables new user registration by capturing essential user details, validating inputs, and communicating registration data to the backend server for account creation. Error handling mechanisms and feedback mechanisms enhance user interaction and streamline account management, promoting user engagement and satisfaction.



PRODUCT DEMO

Vehicle Registration

OVERVIEW OF VEHICLE REGISTRATION PROCESS

Vehicle registration in the app enables users to add new vehicles to their insurance policy seamlessly. RegisterVehicleActivity captures vehicle details such as type, number, and owner name through user-friendly input forms, validates data entries, and submits information to the backend server via RESTful APIs for storage. The process enhances user convenience by simplifying vehicle management and insurance policy updates, facilitating real-time data synchronization and ensuring accurate record-keeping within the app.

VEHICLE INSURANCE APP

UI/UX Design

UI/UX design in the Vehicle Insurance App prioritizes intuitive navigation, visual clarity, and user engagement. MainActivity and SignUpActivity feature streamlined input forms with error handling mechanisms, ensuring a user-friendly registration and login experience. HomePageActivity displays personalized user information, registered vehicle details, and interactive buttons for seamless navigation to additional app functionalities. The use of CardViews, Toolbars, and ScrollView enhances visual appeal and functionality, optimizing screen utilization and promoting a cohesive app interface across diverse Android devices.



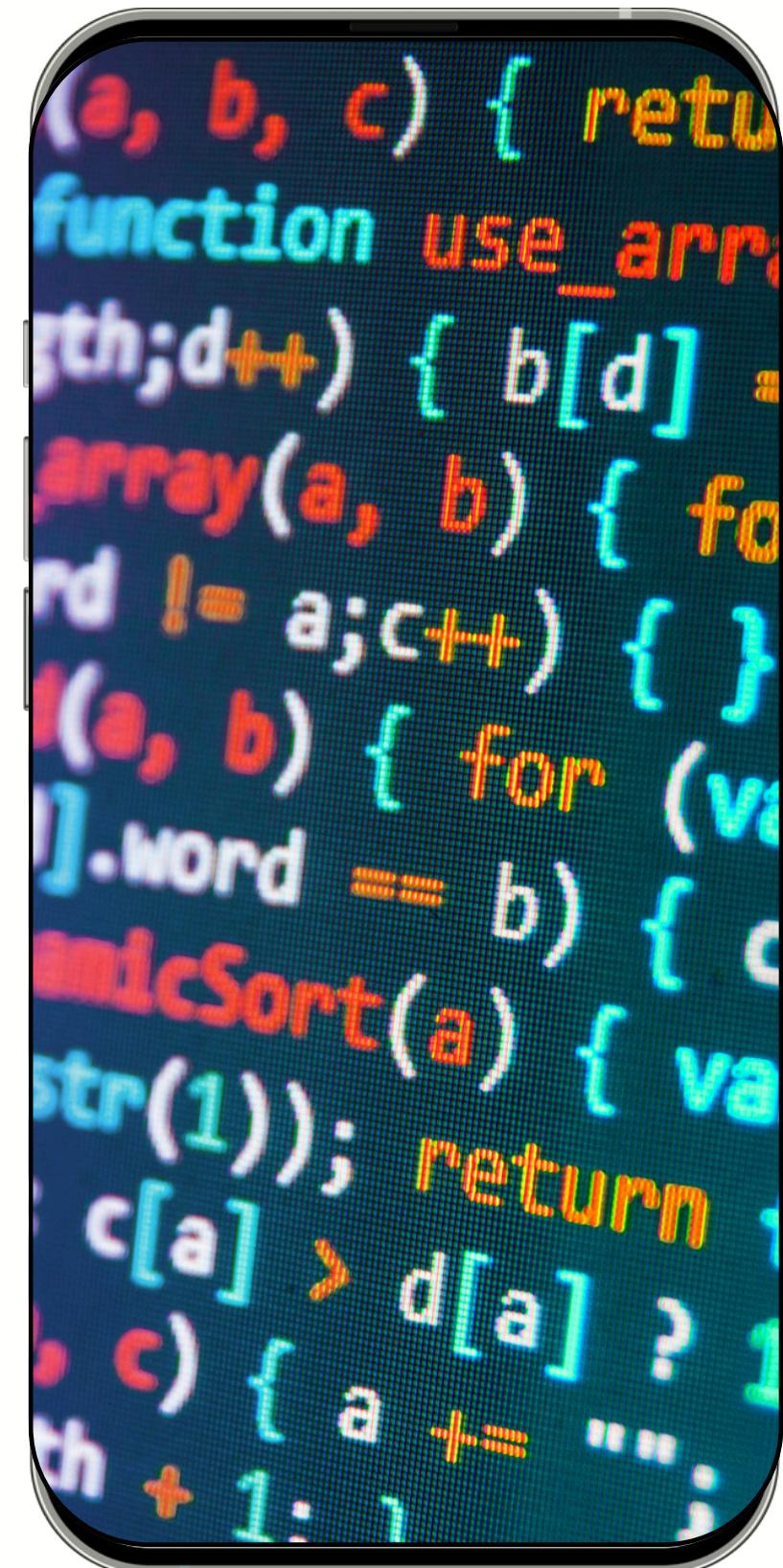
VEHICLE INSURANCE APP

Testing and Debugging

Rigorous testing ensures the reliability, functionality, and performance of the Vehicle Insurance App across various Android devices and operating system versions. Testing methodologies include unit testing, integration testing, and user acceptance testing (UAT), verifying app functionalities such as user authentication, vehicle registration, claim submission, and policy retrieval. Debugging techniques address issues related to UI rendering, network connectivity, backend integration, and data synchronization, ensuring a seamless user experience and adherence to quality standards throughout app development.

Tested using the thunder client

michaeljoshua.p1504@gmail.com



Vehicle Insurance App

VEHICLE INSURANCE APP

Challenges and Solutions

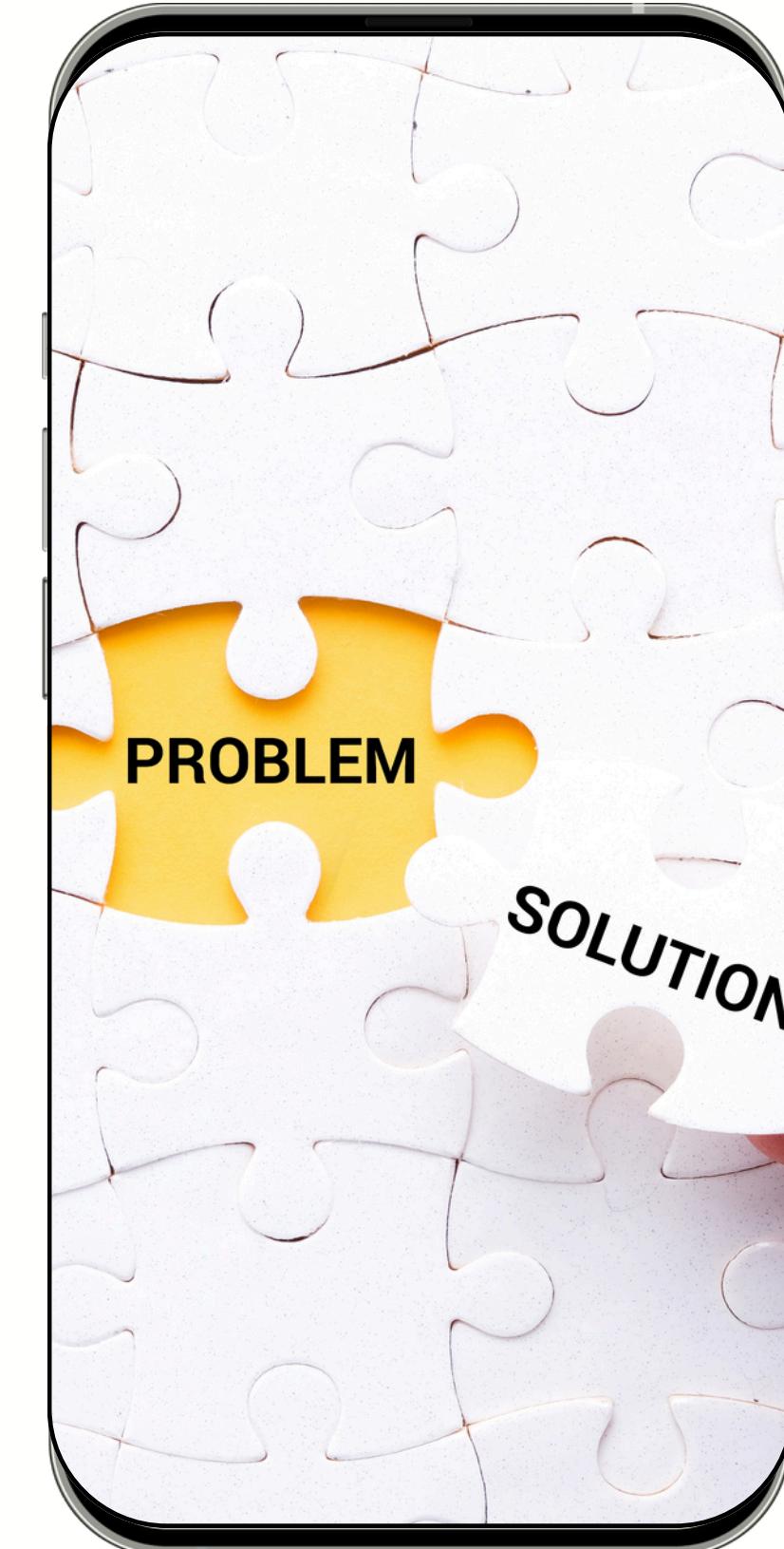
- The development of the Vehicle Insurance App encountered challenges such as backend integration complexities, data synchronization issues between frontend and backend components, and UI/UX design optimization across diverse Android devices. Solutions implemented include modularizing backend APIs for scalability, implementing data caching strategies for offline functionality, and conducting comprehensive UI testing for consistent user experience. Collaboration among cross-functional teams and agile project management methodologies facilitated timely issue resolution and continuous improvement, enhancing app robustness and performance.



VEHICLE INSURANCE APP

Challenges and Solutions

- The development of the Vehicle Insurance App encountered challenges such as backend integration complexities, data synchronization issues between frontend and backend components, and UI/UX design optimization across diverse Android devices. Solutions implemented include modularizing backend APIs for scalability, implementing data caching strategies for offline functionality, and conducting comprehensive UI testing for consistent user experience. Collaboration among cross-functional teams and agile project management methodologies facilitated timely issue resolution and continuous improvement, enhancing app robustness and performance.



Conclusion and Future Work

The Vehicle Insurance App represents a comprehensive solution for managing vehicle insurance claims, policy information, and user interactions through a mobile platform.

Future enhancements may include photo claim processing and agent communication.

GitHub Links

Front-end: <https://github.com/Michaeljoshua1504/VehicleInsuranceClaimApp>

Back-end: <https://github.com/Michaeljoshua1504/backendvehicleinsurance>

Michael Joshua

Presented By

MOBILE DEVELOPMENT

michaeljoshua.p1504@gmail.com

Thank You