



Access Online



Article DOI:

10.5281/zenodo.21408535

¹ Student, Dept. of
ECE, Sharnbasva
University Exclusively
for women
Kalaburgi, India

² Professor, Dept. of
ECE, Sharnbasva
University Exclusively
for Women
Kalaburgi, India

How to Cite:

Vrishti & Dr. Basanti
Ghanti (2026),
Mobile Application
Development for
Regional Transport
Office (Rto) Services
using Python,
International
Educational Journal of
Science & Engineering
(IEJSE), Vol: 9, Special
Issue, May 2026
194-197

MOBILE APPLICATION DEVELOPMENT FOR REGIONAL TRANSPORT OFFICE (RTO) SERVICES USING PYTHON

Vrishti¹, Dr. Basanti Ghanti²

ABSTRACT

Transport management departments handle large amounts of citizen data related to vehicles, driving licenses, and traffic violations. Traditional manual processes for managing these records take a lot of time and are prone to mistakes. This paper outlines the design and development of a mobile application for Regional Transport Office (RTO) services in Karnataka. It is built using Python and the Kivy framework. The application offers role-based access for administrators, officers, and citizens. Users can manage vehicle registrations, view and pay traffic challans, and check the status of their driving licenses. The system features an easy-to-use graphical interface, search and filter options, and real-time payment processing within the app. The application simplifies citizen-facing transport services and shows how Python-based mobile development can effectively enhance e-governance solutions.

KEYWORDS: Mobile Application, Kivy Framework, Python, RTO Karnataka, Traffic Challan, Vehicle Registration

I. INTRODUCTION

The main goal of this project is to develop a mobile application for RTO services in Karnataka using the Python programming language. The RTO office manages vehicle registrations, driving licenses, and traffic challans. Handling all these records manually takes a lot of time and can lead to errors. A mobile application allows citizens to access these services easily from their phones. The app was built with Kivy, an open-source Python library for creating mobile apps. It works on Android, iOS, and desktop platforms. The app has login-based access for three roles. Depending on the role, different screens and options are available to the user. This project helped in understanding how Python can be used to create real-world mobile applications for government purposes.

II. LITERATURE SURVEY

Kivy and Python have been successfully utilized in the development of cross-

platform mobile apps by implementing an appropriate framework and tools. Kivy offers the advantage of the same codebase being able to run across all platforms. Evidence indicates that both Python and Kivy provide flexibility and ease of development for the creation of e-government applications; therefore, they are a viable option for developing applications for use by the government. One of the most common themes identified in research on e-government apps is that they can provide a wide range of features, such as searching, filtering based on status, and the ability to pay bills online all of which make e-government apps more useful.

Previous research conducted in transportation management in India has indicated that a high portion of Government web portals are not designed for mobile devices. Citizens will experience added difficulty accessing vehicle records and paying fines on their

mobile device because of the small size of the device screen. Multiple studies have proposed the use of role-based access control as an essential feature for any application that contains personally identifiable information belonging to citizens. Designing the application to incorporate these recommendations has been a major consideration in the design process for this project.

III. SYSTEM DESIGN

The objective of this project is to create a mobile application that allows people in Karnataka State to access RTO services online by using the Python Programming Language. The RTO office is responsible for operating vehicle registration, issuing driving licenses and traffic challans. Managing these records in a manual way requires a significant amount of time and may result in recordkeeping errors. A mobile application provides citizens with an alternative method for obtaining these services through their mobile device. The mobile application was developed using Kivy, an open-source software library that enables developers to build mobile applications using the Python programming language. The mobile application can be deployed on Android, iPhone, and also desktop computers. There are three types of users (roles) so each has access to different screens and options based upon their assigned role. Completing this project served as a means to better understand how to utilize Python to build real-world mobile Government Applications.

IV. METHODOLOGY

Using Kivy and Python 3, the project was developed incrementally, beginning with the login page and functioning through each feature screen in turn. Code level constructs allowed for all UI element to be developed in code rather than using KV language files. This approach enables conditional inclusion of elements on the screen, and allows for layout changes depending upon the type of user accessing the application.

To keep the code clean and eliminate duplication, there are helper functions. The `lbl()` function creates label widgets while the `make_btn()` function creates styled buttons with rounded corners. Lastly, the `card_box()` function creates white card layouts for all list screens. To ensure all screens have a consistent

color theme, there was a dictionary of colors used throughout the app. Green represents an active/paid status, red represents an expired/unpaid state, orange represents a challan and purple represents a licence.

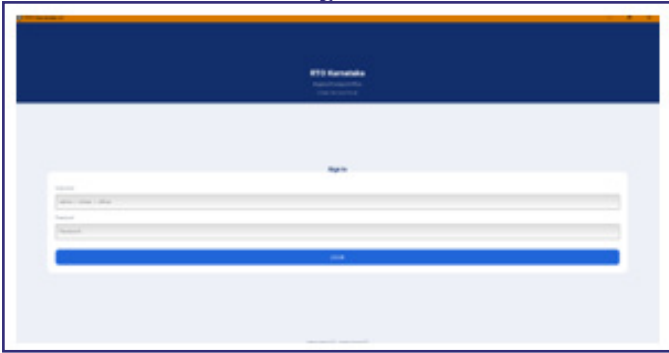
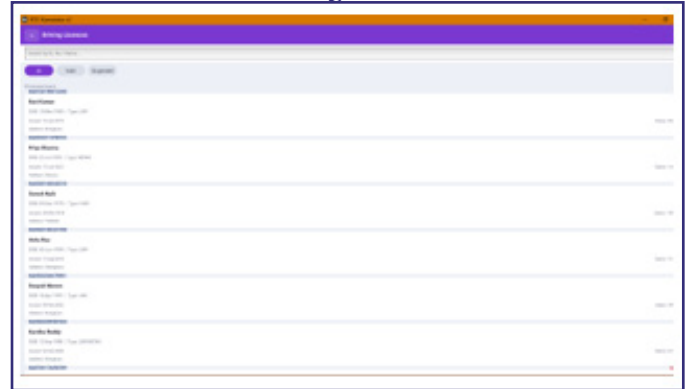
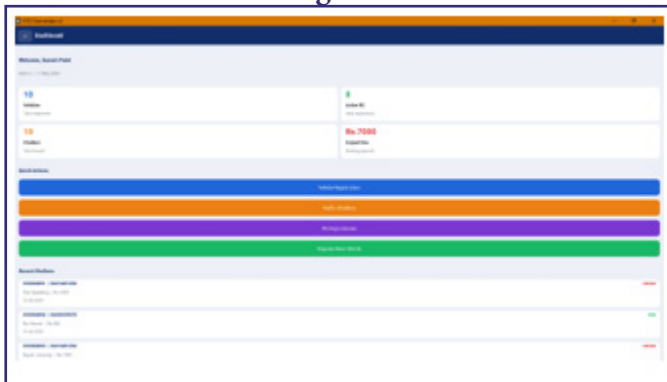
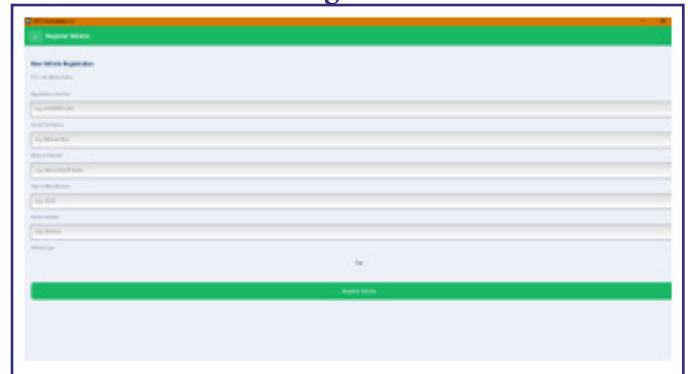
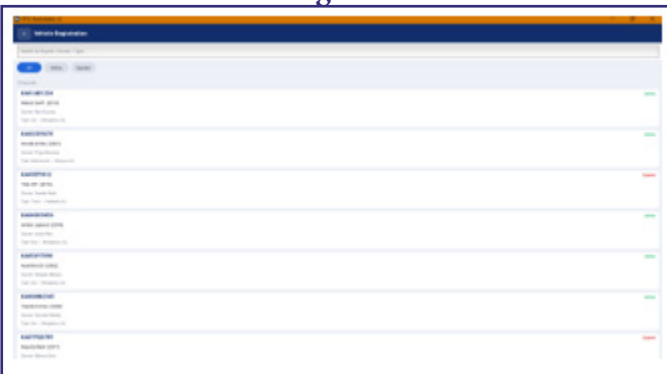
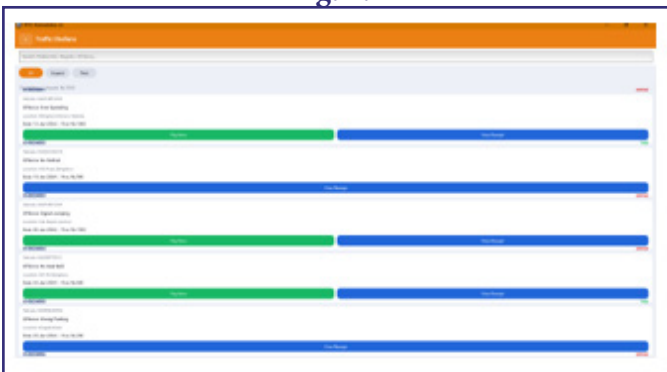
Searches are real-time in nature, updating the list automatically as the user types without requiring a button press. Additionally, filter chips can be used on every screen to display only Active, Expired, Paid, Unpaid, Valid or Suspended records depending on the particular screen being viewed. During the course of development there were a number of issues encountered with sizing of layouts and binding of widgets which furthered understanding about how Kivy manages updates to the user interface.

V. RESULTS AND DISCUSSIONS

They've conducted testing of the applications using simulated screen size as that of what is considered to be a baseline Android phone. All 7 screens perform functions as expected. The login screen validates the user's credentials and directs them to their dashboard depending on their role. The dashboard displays several summary cards including the total number of vehicles, number of active registrations, total number of charges, and the amount due on those charges.

The vehicles page contains 10 sample records and the search and filter functionality are functioning as expected. There are also 10 records of violations displayed on the "Challan" page. When a user selects "pay now", the Challan is marked as Paid, and the results are updated immediately on the screen. The receipt page displays a formatted Challan receipt similar to an official document from the perspective of being a receipt that a police officer would give out if they stopped someone for a traffic violation and had issued them a citation.

The Licenses page includes 8 records. Each of the records may have a either a status of "valid" or "suspended". The Add Vehicle form used to add vehicles has validation checks in place to ensure there is no empty fields and validation checks to ensure there are not duplicate registration numbers before saving the new vehicle record.

Fig.1.1**Fig.1.5****Fig. 1.2****Fig. 1.6****Fig. 1.3****Fig. 1.4**

VI. CONCLUSION

This project has been helpful in learning how to create a mobile app using Python with Kivy, as well as verifying that the RTO app has a good working interface for vehicle registration/rto based vehicle challan payments and checking driving license information. Role based user login will provide a means to control which type of user can see or do what within the app. I gained a lot of valuable experience in mobile UI development, event handling, and app development with real world use cases while working on this project.

The helper function method helped keep my code well organized and simple to add additional functionality to my app. Additionally I used a reference color dictionary throughout the entire app which maintained a uniform appearance of my entire application. I plan to integrate a real database for storing user and payment information into my future version of the app. I also plan to package the application as an Android APK for use by real citizens..

VII. FUTURE SCOPE

By incorporating a live back-end database like SQLite or a REST API, it will maximize the potential of the app. In addition, using a real payment gateway like UPI will replace the current simulation of payments. Users may be able to set reminders for their renewal due dates and/or fines via push notifications.

To add functionality for all citizens in Karnataka, the app should also contain support for the Kannada language. Document upload functionality for insurance and PUC certificates would also be beneficial. Finally, the app can be packaged as an Android APK with Buildozer, giving all citizens access to download and use the app.

REFERENCES

1. M. Virbhadrar, A. J. Patel, "Python Kivy Based Mobile Applications for E-Government", International Journal of Computer Applications, vol. 183 no. 12, 2021, pp. 1-6.
2. S. Chandra, R. K. Singh, "A Comparative Study On Cross-Platform Mobile Applications Development", Journal of Software Engineering Research, vol. 9 no. 2, 2022, pp. 45-52.
3. The Government of Karnataka, "Motor Vehicles Rules Karnataka", Dept of Transport Govt of Karnataka, Bengaluru 2023. [Online]. Available: <https://transport.karnataka.gov.in/>
4. A. Noel, B. Thomas, "Role Based Access in m-government Applications", International Journal of Advanced Computer Science, vol. 13 no. 4, 2022, pp. 88-95.
5. Kivy Documentation, "Kivy Framework Documentation", Kivy Organisation, 2023 [Online]. Available: <https://kivy.org/>
6. P. Ramachandran, V. Iyer, "Digitally Transforming the Indian Transportation Sector", Journal of Public Administration, vol. 14 no. 1, 2022, pp. 12-20.
7. L. Garofalo, Python Mobile Development using Kivy, Packt Publishing, 2020, Birmingham, United Kingdom.
8. Python documentation, [Online]. Available: <https://docs.python.org/3/>
9. R. Kamath, S. Nair, "Design Citizen Centered E-Government Portals for India", in Proceedings - International Conference on E-Governance, New Delhi, 2021, pp. 102-108.