A report on project - **ReachOut**

ABSTRACT

This project report discusses the complete design as well as implementation for **ReachOut**, a one-stop social media aggregate platform built on Django. The main goal of the system is to simplify users' digital footprint by merging various social media profiles — including Instagram, LinkedIn, Snapchat, Reddit, Facebook and Twitter — into one, easily shareable platform. You are curated on your profile links which are dynamic, secure and aesthetic and you can manage and showcase your connected accounts in the dashboard.

The system's functionalities encompass: user authentication , UI design that adapts to smaller screens, a backend that interacts with the data store through Django's ORM and logic that integrates the profile seamlessly. Also, future enhancements are planned to add a Flutter-based mobile interface to support this system. The project follows modular principles, enabling code to be reused, while the user's data is also stored and managed in a privacy-conscious way; the user's data is not stored except for linking accounts.

We developed the lifecycle in a systematic way, utilising components from the **Agile methodologies**, layered with Interative peer review and academic mentorship. The project was spread out over a span of three months and consisted of detailed requirement analysis, system design, implementation, testing, and finally deployment.

TABLE OF CONTENT

CHAPTER 1 INTRODUCTION

1.1 PROJECT STATEMENT

Web based designed solution for users to connect their friends socials everywhere. ReachOut allows users to create unique QR code for their profile and share it with their friends or strangers so they can scan qr code and get access to all social media connected to qr. It includes secure login register of users, making the platform more secure and reliable. Reachout scales the network for socials.

1.2 PURPOSE

The primary purpose of ReachOut project is to provide seamless solution for users who frequently got engaged into social medias. Instead of manually connecting the socials, ReachOut aims to provide a public customizable page regarding their social presence.

1.3 OBJECTIVE

To build a secure and user-friendly E-Voting platform using Django.To allow administrators to manage elections (voters, candidates, positions) with ease.To enable voters to register, log in, and vote securely online.To automate vote tallying and display results using visual charts.To support result printing in PDF format for record-keeping and transparency.

1.4 SCOPE

- Allow them to link their social media profiles for Instagram, LinkedIn, Facebook, etc.
- Offer an easy to use dashboard for your profile links to be added, edited, and removed.
- Create your own custom, reusable public page with all your user links.
- It should acceptable in various devices and browsers.
- Future plans has the integration of the mobile app through Flutter Also, need to add the analytics of the clicked links.

1.5 TECHNOLOGY AND LITERATURE REVIEW

This project is built using Django - a high level Python Web framework focused on clean code and security. The built-in components of Django, like authentication, ORM, and templating system, make it easier to create scalable and secure web applications. Frontend is styled with ReactJs, providing a responsive and modern user interface. Will integrate with Flutter in the future for mobile access.

However, studies and industry trends show that link aggregation tools are a major game-changer when it comes to improving digital visibility and simplifying content sharing, particularly for content creators, influencers, and professionals. "Similar platforms such as Linktree and Carrd exist, but ReachOut sets itself apart with additional customization, future extensibility and developer-oriented, privacy-focused approach.

CHAPTER 2 PROJECT MANAGEMENT

2.1 PROJECT PLANNING

2.1.1 Project Development Approach and Justification

The ReachOut was developed adopting a gradual, modular method inspired by the Agile model. The work was split into small tasks and phases and included regular testing and feature merging after each phase. A product that slowly, but surely improved through feedback loops and iterations.

The UI/UX was designed with Figma and developed with React. js for the web frontend. The choice to use React was made based on its component-based architecture, high speed and ease of manipulation with interactive elements. Backend — Django Flutter(for mobile app — cross platform) Develop and test each feature in isolation — profile linking, QR code generation, analytics, et cetera — in turn before advancing to the next.

2.1.2 Project Effort and Time, Cost Estimation

The project is still ongoing, but the web version has been completed, and mobile app development is in progress. The expected total timeline for the complete system (web + mobile) is about 4 to 5 months.

Assuming 15–20 hours of work per week:

Table 2.1 Project cost estimation

Resource	Estimation
Development Time	240 hours
Project Duration	4 months
Estimated Cost	No external costs incurred

2.1.3 Roles and Responsibilities

Since the project was individually developed, all roles were handled by a single contributor

Table 2.2 Role and responsibilities

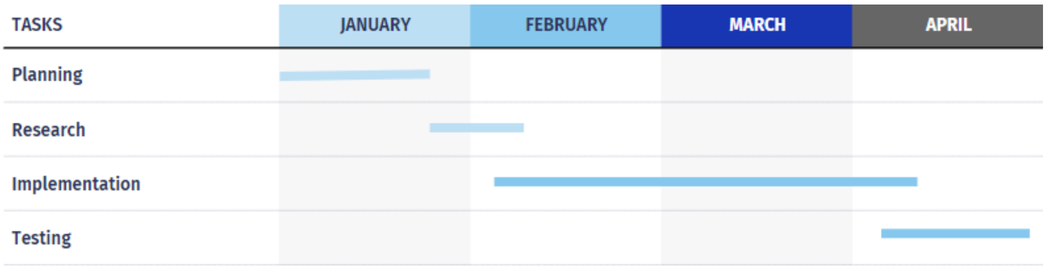
Role	Responsibility
Project Manager	Planning, scheduling, monitoring
Developer	System design, coding, testing, deployment
UI/UX Designer	Front-end design and integration with AdminLTE
Tester	Manual testing, validation of use-cases and bugs
Documentation Lead	Writing reports, user guides, and project documents

2.1.4 Group Dependencies

This project was self-contained and had no external group dependencies. However, it made use of open-source tools and libraries such as:

- Django (framework)
- ReactJs (Frontend)
- PostgresDB (Database)
- Python libraries like reportlab or xhtml2pdf for PDF generation

2.2 PROJECT SCHEDULE



CHAPTER 3 SYSTEM REQUIREMENTS STUDY

3.1 USER CHARACTERISTICS

The system is designed for two primary types of users.

Table 3.1 Characteristics of users

User Type	Description
Administrator	Handles social platform developer account. Validates user’s linked social profile. Manages Users and their actions.
User	Link socials, Generate sharable qr, shares with friends, follow in-platform other users. Find their contact’s socials.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.2.1 Minimum Hardware Requirements

Table 3.2 Hardware Requirements

Component	Specification
Processor	Dual-core 2.0 GHz or higher
RAM	4 GB or higher
Storage	At least 500 MB free disk space
Display	1024x768 resolution or higher
Internet	Required for accessing the system

3.2.2 Software Requirements

Table 3.3 Software Requirements

Software	Version/Details
Operating System	Windows 10/Linux/macOS
Python	3.8 or above
Django	3.2 or above
Database	SQLite (default), can be extended to PostgreSQL/MySQL
Web Browser	Google Chrome, Firefox, or Microsoft Edge

3.3 ASSUMPTIONS AND DEPENDENCIES

3.3.1 Assumptions

- Users Use a modern smart phone/device/computer with internet facility.
- The users know how to use social media properly and can correctly direct to their own accounts.
- No manual admin approval is needed, as each user will register and manage their own profile.
- The analytics feature is a premium service, writable only after payment, and as such users know it is not available until they have paid for it.
- Once the app is published, mobile users will need to download the app from official app stores.
- This assumes users are engaging in a shared context that would allow scanning (e.g., events, meetings).

3.3.2 Dependencies

- React needs to be configured and installed properly to run a web application. js (frontend) and Django (back end).
- The mobile application is built using the Flutter SDK and needs to be compiled for Android and iOS platforms.
- This is necessary when syncing user data and retrieving analytics, as well as rendering linked social profiles.
- Graphic design relies on QR code generator libraries are added into web and mobile versions.

CHAPTER 4 SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

Today, users manage and share multiple social media profiles by copying and pasting individual links or using third-party services that rarely offer customization, branding, or analytics. There is no standard way to represent a personal digital identity or monitor interactions with shared links. Influencers, Professionals, and Creators who rely on having a clean and productive digital footprint are certainly hindered by these constraints.

4.2 PROBLEMS AND WEAKNESSES OF THE CURRENT SYSTEM

- No branding or personalization options in existing link-sharing tools
- Most alternatives lack mobile and offline compatibility
- Lack of native insights to see profile views or engagement
- No branding or personalization options in existing link-sharing tools
- Susceptibility to manipulation or fraud
- Difficult to scale or audit

4.3 REQUIREMENTS OF THE NEW SYSTEM

4.3.1 Functional Requirements

- Secure login and registration for user authentication
- Dynamic profile creation with links to social media
- QR Code generation distinct per user
- Personal brand and link visibility management dashboard
- Analytics paid module that include:
  - Region-wise visitor data
  - Link clicks and like counts
  - Visual charts to gain insights into data
  - Candidates
- Web frontend built in React. js, mobile version with Flutter

4.3.2 Non-Functional Requirements

- **Security:** User data is stored securely, validating and authorizing on backend.
- **Usability:** Minimalistic interlaced UI/UX designed on Figma with solo smooth navigation process.
- **Availability:** Hosted on cloud servers for 24/7 uptime with web and mobile access
- **Scalability:** Architecture can handle an increasing number of requests without significant performance issues.
- **Maintainability:** with dedicated user account, analysis, front end
- **Auditability:** You provide legal and platform permissions for each integrated social network to help keep data safe.

4.4 FEASIBILITY STUDY

4.4.1 Contribution to Organizational Objectives

People use ReachOut to easily manage and share their online identities, making it especially valuable to professionals, creators, and influencers. It brings together convenience, data insight and privacy: features that are all important to users looking for personalized experiences and expanded reach—and modern networking just like traditional shaping.

4.4.2 Technical & Schedule Feasibility

The system has been built with two of the most popular technologies —Django for the backend, React. js in the web frontend, and Flutter for mobile. The tools enabled us to finish the MVP under 3 months addition in the mobile version are in incremental stages.

4.4.3 System Integration Feasibility

It can be integrated with:

- Social media platform external apis (e.g: Instagram, LinkedIn)
- PostgreSQL or Firebase (cloud databases)
- Email/SMS APIs for notifications or OTP
- Database systems like PostgreSQL or MySQL

4.5 ACTIVITY/PROCESS IN NEW SYSTEM

- User registration and profile creation
- Account and page links on various social media platforms.
- A unique QR code gets generated and saved
- User can see profile preview and share it
- Analytics and chart-based reports are available to paid users
- Admin reviews usage on the platform and manages any policy/feature changes

4.6 FEATURES OF THE NEW SYSTEM

- Access to web and mobile app versions
- All social media handles in one personalized profile
- Each user has their own QR code generated
- Chart and regional stats paid analytics dashboard
- Secure user authentication
- React based lightweight, responsive design js and Flutter
- Behind the scenes, new data is continuously updated in real time with backend services.
- Compliance with social platforms data privacy
- Shared easily on multiple platforms

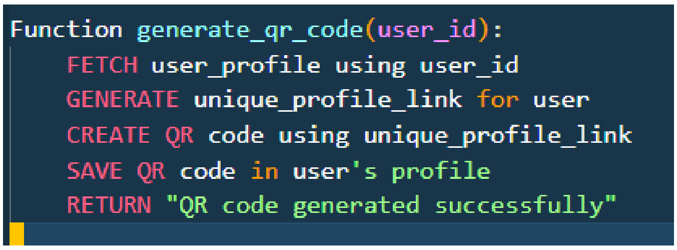
CHAPTER 5 SYSTEM DESIGN

5.1 SYSTEM APPLICATION DESIGN

The ReachOut Logic is designed using the Object-Oriented Design (OOD) paradigm with Django's Model-View-Template (MVT) architecture. It ensures modularity, reusability, and scalability while separating the logic, database models, and user interface.

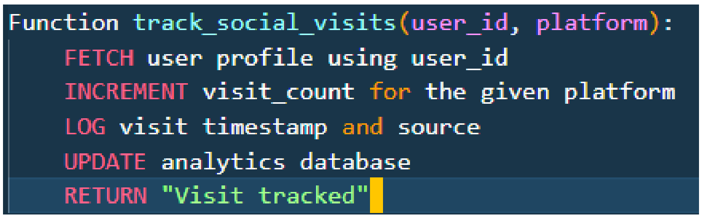
5.1.1 Method Pseudocode

1. **Generate QR code**



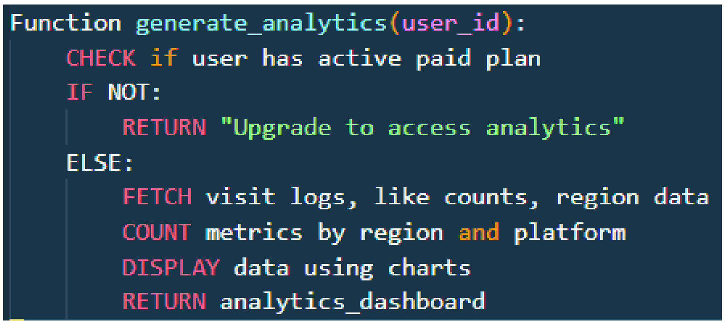
```
Function generate_qr_code(user_id):
    FETCH user_profile using user_id
    GENERATE unique_profile_link for user
    CREATE QR code using unique_profile_link
    SAVE QR code in user's profile
    RETURN "QR code generated successfully"
```

2. **Track social visits**



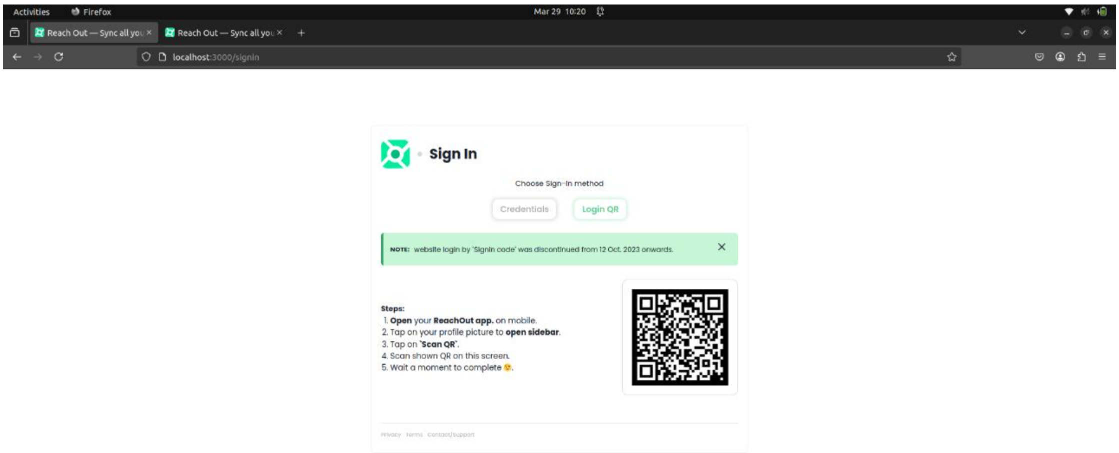
```
Function track_social_visits(user_id, platform):
    FETCH user profile using user_id
    INCREMENT visit_count for the given platform
    LOG visit timestamp and source
    UPDATE analytics database
    RETURN "Visit tracked"
```

3. **Generate analytics**



```
Function generate_analytics(user_id):
    CHECK if user has active paid plan
    IF NOT:
        RETURN "Upgrade to access analytics"
    ELSE:
        FETCH visit logs, like counts, region data
        COUNT metrics by region and platform
        DISPLAY data using charts
        RETURN analytics_dashboard
```

5.1.2 Screenshots

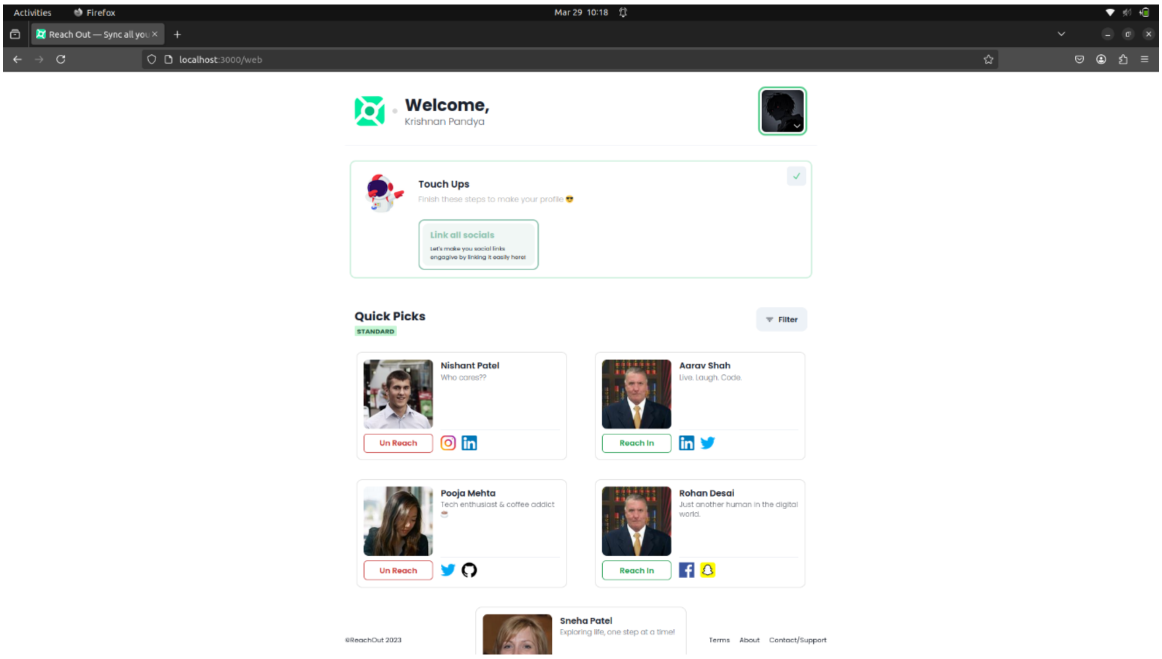


Choose Sign-in method

NOTE: website login by `SignIn code` was discontinued from 12 Oct. 2023 onwards.

Steps:

1. **Open** your **ReachOut app.** on mobile.
2. Tap on your profile picture to **open sidebar**.
3. Tap on `**Scan QR**`.
4. Scan shown QR on this screen.
5. Wait a moment to complete 😉.



Finish these steps to make your profile 😎

Let's make you social links engagive by linking it easily here!

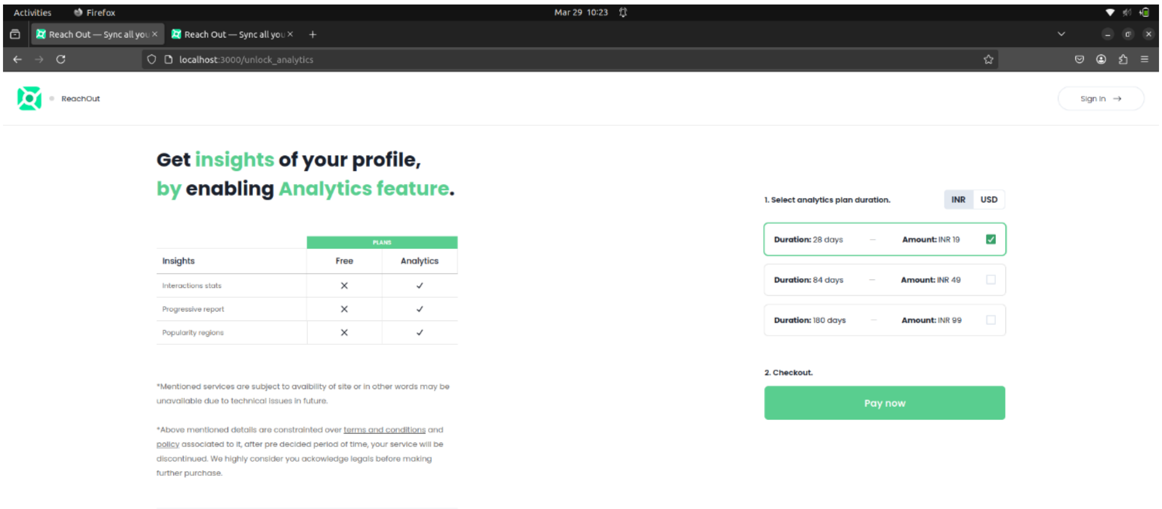
Who cares??

Live. Laugh. Code.

Tech enthusiast & coffee addict ☕

Just another human in the digital world.

Exploring life, one step at a time!



Get insights of your profile, by enabling Analytics feature.

	PLANS	
Insights	Free	Analytics
Interactions stats	✕	✓
Progressive report	✕	✓
Popularity regions	✕	✓

*Mentioned services are subject to availbility of site or in other words may be unavailable due to technical issues in future.

*Above mentioned details are constrainted over terms and conditions and policy associated to it, after pre decided period of time, your service will be discontinued. We highly consider you ackowledge legals before making further purchase.

1. Select analytics plan duration. INR USD

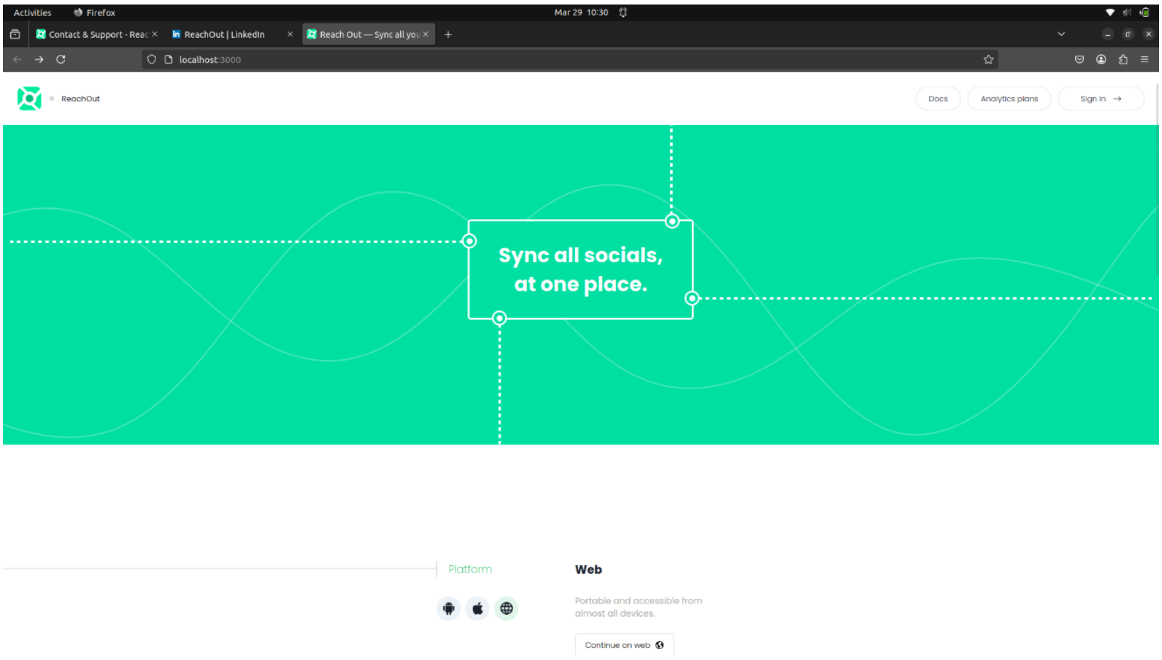
Duration: 28 days — **Amount:** INR 19 ☑

Duration: 84 days — **Amount:** INR 49 □

Duration: 180 days — **Amount:** INR 99 □

2. Checkout.

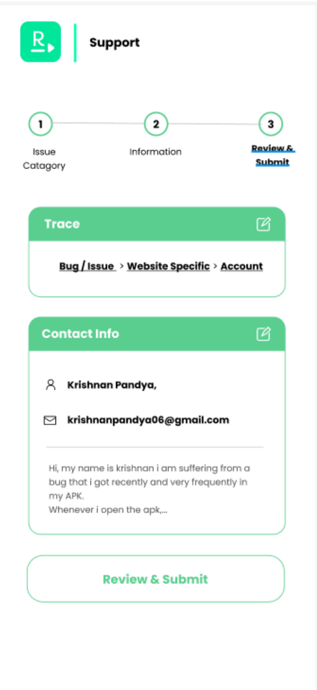
Pay now



Web

Portable and accessible from almost all devices.

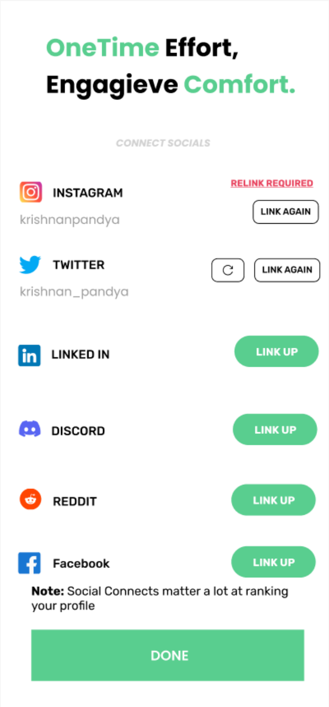
Continue on web



Krishnan Pandya,

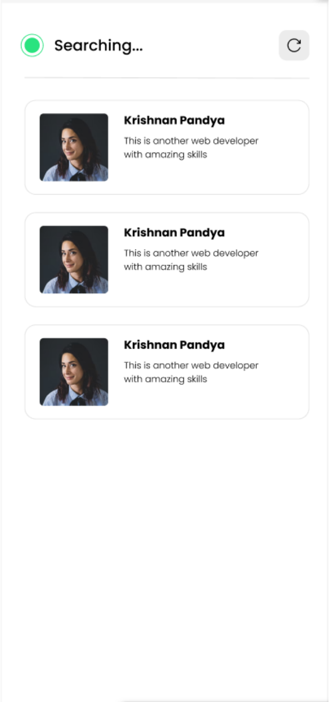
krishnanpandya06@gmail.com

Hi, my name is krishnan i am suffering from a bug that i got recently and very frequently in my APK.
Whenever i open the apk,...



OneTime Effort, Engagieve Comfort.

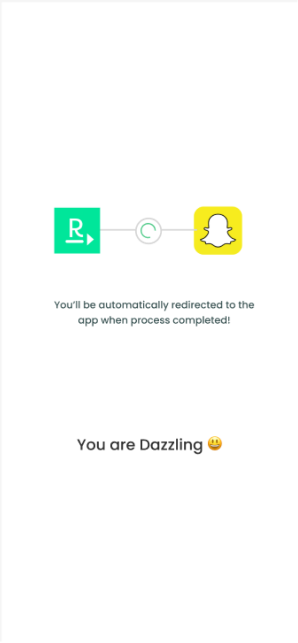
Note: Social Connects matter a lot at ranking your profile



Krishnan Pandya
This is another web developer with amazing skills

Krishnan Pandya
This is another web developer with amazing skills

Krishnan Pandya
This is another web developer with amazing skills



You'll be automatically redirected to the app when process completed!

You are Dazzling 😃

5.2 INPUT / OUTPUT AND INTERFACE DESIGN

ReachOut is a smooth platform with an easy user experience. This gives users their all social media links in one place, provides analytics, and users can easily share their profiles with QR codes. Admin users (project maintainers): They can access platform management tools, and analytics configuration.

5.2.1 Input/ Output

Inputs:

- Data on user registrations (email, password)
- Social media handles added by user, (Instagram, LinkedIn, Twitter, etc.)
- Scan QR Code (for visitor tracking)
- Manage developer API keys and legal documents admin inputs
- Username and password (for some forms of sign in)

Outputs:

- All social media handles will be linked on single profile page.
- Unique QR code for each user
- Click analytics (region-wise, platform-wise)
- Graphs depicting user engagement statistics
- Admin dashboard summary (user stats, usage patterns)
- Paid analytics report access

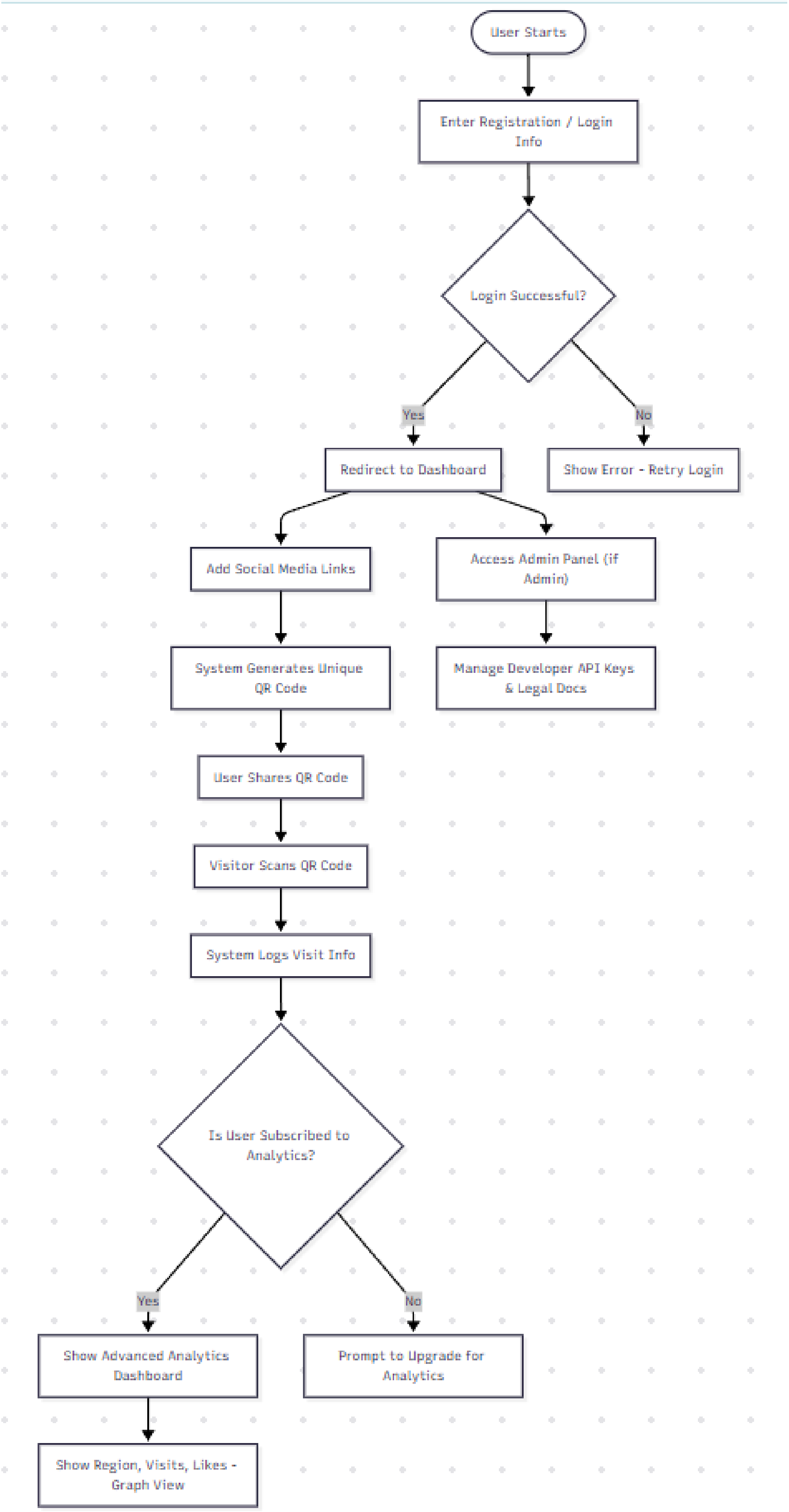

User Starts
Enter Registration / Login Info
Login Successful?
Yes
No
Redirect to Dashboard
Show Error - Retry Login
Add Social Media Links
Access Admin Panel (if Admin)
System Generates Unique QR Code
Manage Developer API Keys & Legal Docs
User Shares QR Code
Visitor Scans QR Code
System Logs Visit Info
Is User Subscribed to Analytics?
Yes
No
Show Advanced Analytics Dashboard
Prompt to Upgrade for Analytics
Show Region, Visits, Likes - Graph View

Fig 5.4 Input Output Design

5.2.2 Interface Design

Admin Panel UI:

- Sidebar navigation: Dashboard, Voters, Candidates, Positions, Results
- CRUD operations via modals or dedicated forms
- Vote results with interactive charts

Voter UI

- Registration and Login page
- Ballot screen showing available positions and candidates
- Confirmation of vote submission

5.2.3 State Transition Diagram

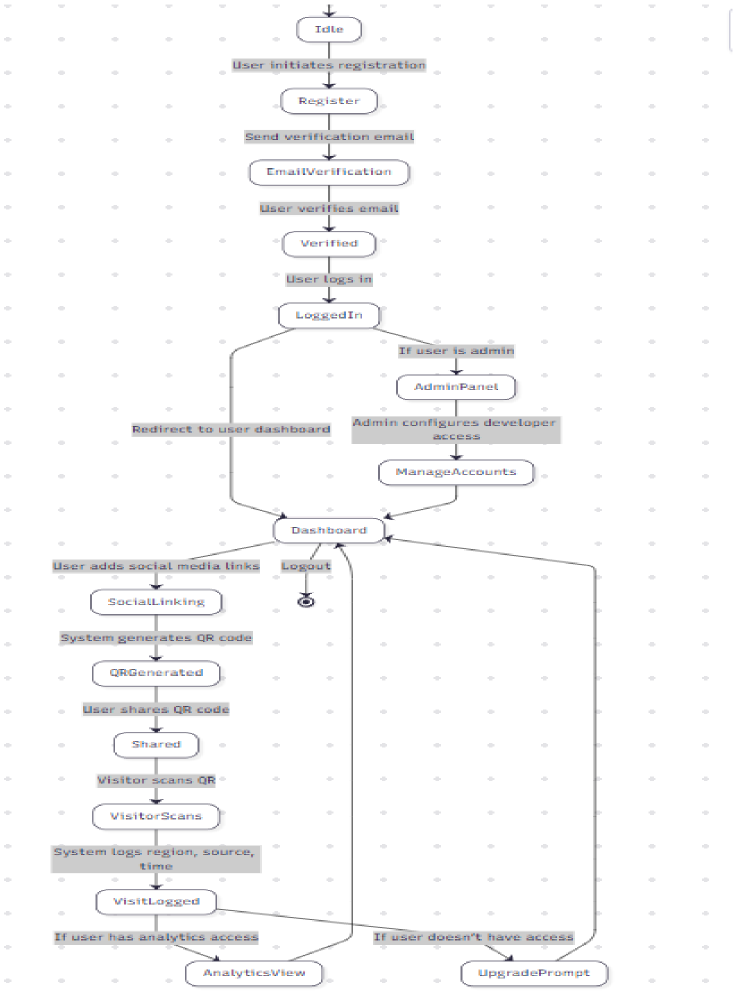


Fig 5.5 Complete State Transition Diagram

5.3 INFORMATION ARCHITECTURE

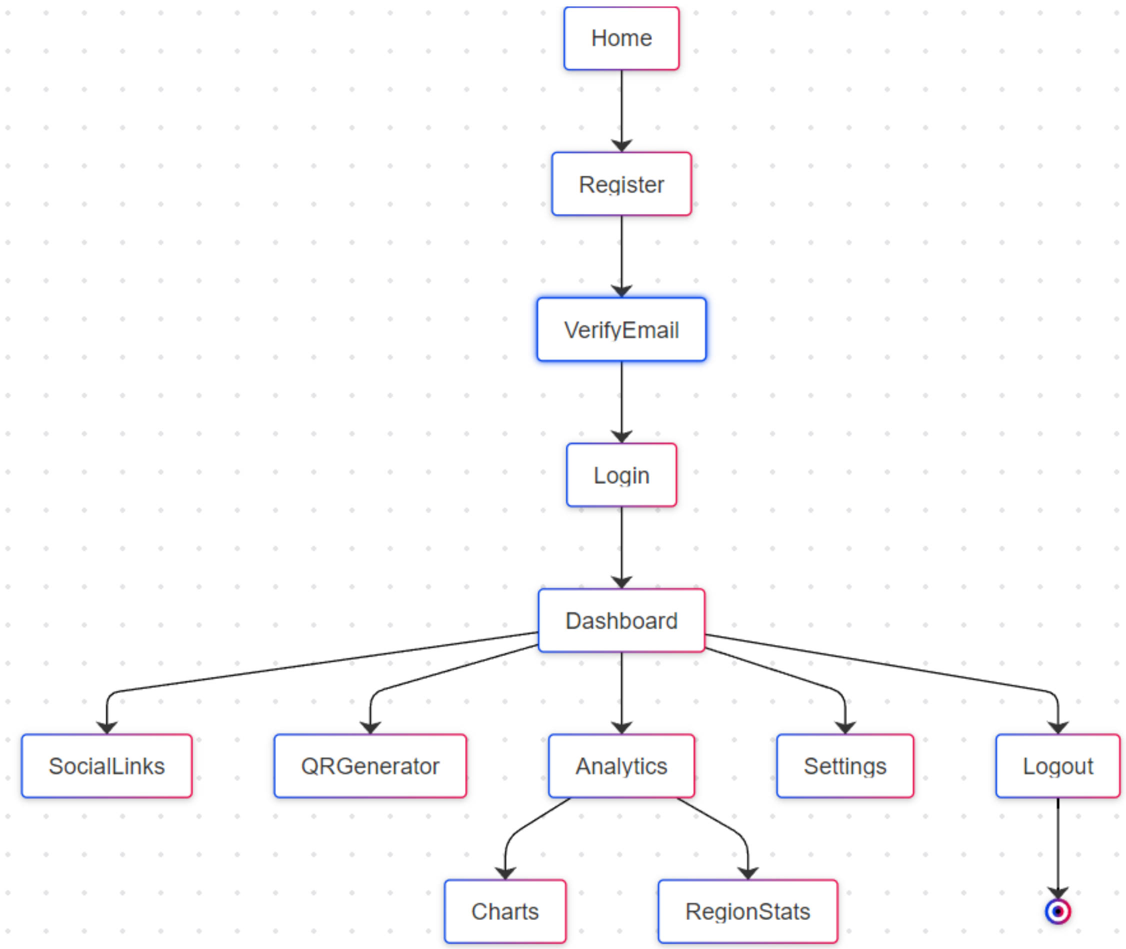


Fig 5.8 Information Architecture

5.4 USER FLOW

5.4.1 User

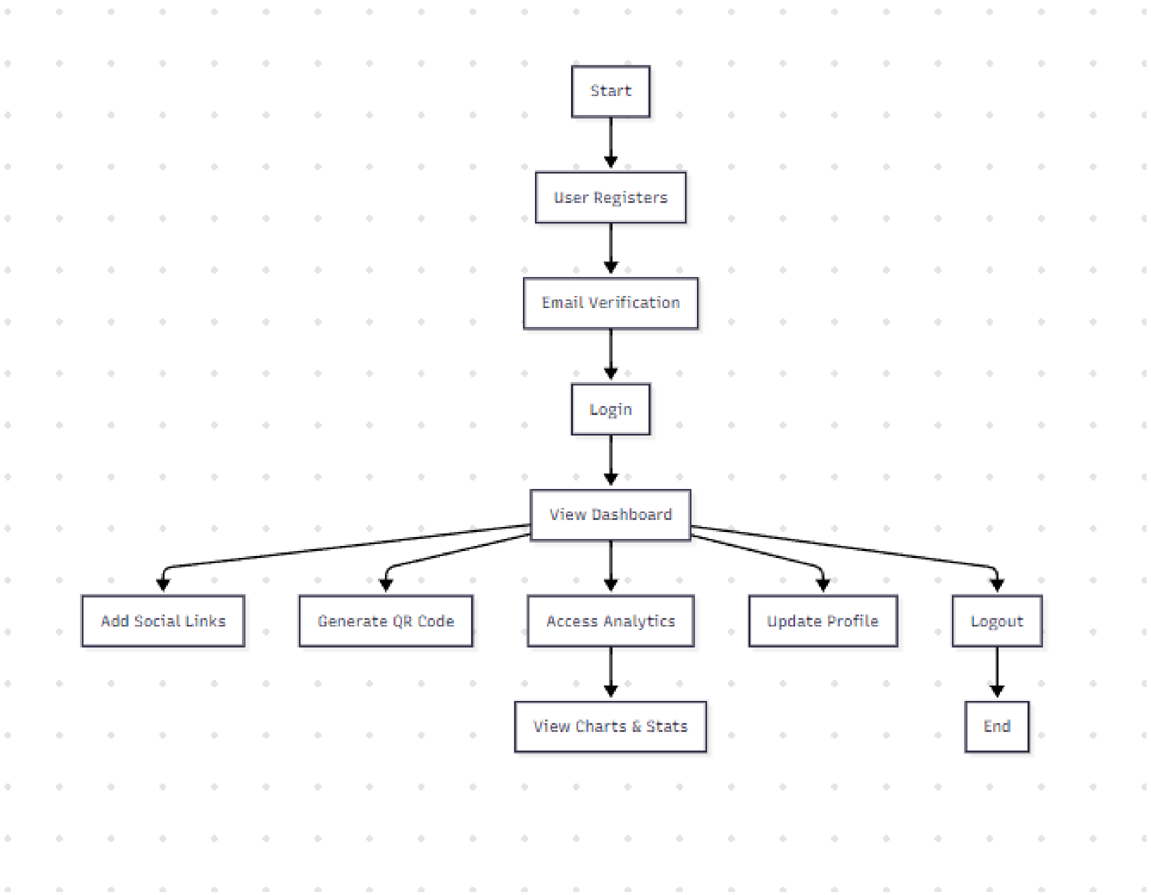


Fig 5.9 User Flow

CHAPTER 6.0 – IMPLEMENTATION PLANNING

6.1 IMPLEMENTATION ENVIRONMENT

ReachOut is built in a single developer environment but designed for multi-users. Train on web and mobile apps data with user sensitive data management and live analytics.

- **Interface Type:** GUI (Graphical User Interface)
- **User Interfaces:** Web-based UI for User.
- **Technology Stack:**
  - Backend: Django (Python)
  - Frontend: ReactJs
  - Database: SQLite (development), can scale to PostgreSQL for production

6.2 PROGRAM / MODULES SPECIFICATION

The system is modular and follows Django's **MVT (Model-View-Template)** architecture. The main modules are as follows

Module	**Description**
Authentication	Handles voter registration, login, session management
User Home	Link their social handles. Create their own public QR code.
Admin Dashboard	Manages the users and their actions. Validated linked socails
Validator	Periodically checks for validity of linked socials by users

Table 6.1 System Modules

6.3 SECURITY FEATURES

The ReachOut system is built around security and privacy. There are several key mechanisms in place:

- User Authentication: User and admins must log in before accessing any features.
- JWT: Authentication: For API interaction
- Secure QR code linking: Every individual profile is guarded and recognized uniquely through authenticated sessions.
- Data Transmission: Encrypted via HTTPS, with correct CORS and rate-limiting
- Form Validation: Input Sanitization on Server and Client Side
- CSRF Protection: Django's built-in CSRF tokens used on all forms.

6.4 CODING STANDARDS

Common Practices for structuring your code-base:

- Naming of Class, State And File as Semantic.
- ui and logic written as reusable React and Flutter widgets
- Separation of Concerns: RESTful APIs are used to decouple backend and frontend
- Comments & Docs: Functions and modules are commented well with meaningful comments and docstrings
- It would also be essential to compare how you manage templates, since React uses JSX patterns, whereas Django uses Jingo templating.
- Version Control: All development under the hood is tracked using Git and GitHub
- Figma-Driven Design: All UI/UX design was initially built in Figma and implemented

Chapter 7 TESTING

7.1 TESTING PLAN

7.1.1 Scope

The testing strategy encompasses every functional aspect of the ReachOut platform, verifying its smooth operation in both web and mobile environments:

- User authentication (login, registration, logout)
- User authentication and accessible profile by QR code
- Create a user profile and link a social media profile
- Analytics module (per region stats, likes, visits)
- Security testing (vulnerability scanning, penetration testing, code review)
- Performance testing (load testing, stress testing)

7.1.2 Objectives

- Verify that every module is working correctly and satisfies all criteria.
- Identify bugs in security, logic,UI/UX
- Ensure privacy of user data and secure sharing of QR
- Measurement of the typical vs peak load system's behaviour
- Assess security via token misuse, QR spoofing and injection attempts
- Validate the system's performance under expected load conditions.
- Verify that system is resilient with security penetration testing
- Test to ensure there are adequate test coverage

7.1.3 Test Levels

- Unit Testing: Testing individual functions, classes, and modules.
- Integration Testing: Testing the interaction between different modules and components.
- System Testing: Testing the entire system as a whole.

7.1.4 Testing Resources

- Dedicated testing environment (separate from development and production).
- Test data (realistic user).
- Testing tools: pytest - For running unit and integration tests.
- Testing documentation (test cases, test scripts).

7.1.6 Schedule

Phase	Start Date	End Date
Unit Testing	13-04-2025	17-04-2025
Integration Testing	18-04-2025	20-04-2025
System Testing	20-04-2025	25-04-2025

Table 7 Schedule

7.2 TESTING STRATEGY

7.2.1 Test Types

- Functional Testing: Verifying that the system functions according to the requirements.
  All the functions need to be tested, from the Admin role to the base line users, to test for the security implementation of the site.
- Security Testing: Identifying vulnerabilities and security flaws.
  - Conducted with tools to identify the vulnerabilities. (SQL Injection, XSS, and CSRF).
  - Use of OWASP ZAP
- Performance Testing: Evaluating the system's performance under various load conditions.
  - Metrics are response time, throughput, and resource utilization.
- Usability Testing: Assessing the ease of use and user satisfaction of the system.
  - Conduct surveys to test if the users or admins are easily able to make actions.
- Accessibility Testing: Ensuring that the system is accessible to all users, including those with disabilities.
- Integration Testing: Ensuring that all the components are working correctly by testing the modules or code and validating them.

7.2.2 Test Data

- Valid and invalid voter credentials
- Realistic users data and position data.
- Large datasets to simulate load conditions.
- Malicious input to test security vulnerabilities.

7.3 TEST SUITES DESIGN

7.3.1 Test Suite Structure

Test suites will be organized by module/feature:

- Account Management (Authentication, Registration, Password Management)
- User Management (Create, Read, Update, Delete)
- Analytics Management (Create, Read, Update, Delete)
- Social Link process(Create, Read, Update, Delete)
- QR create process (Create, Read)
- Security Testing
- Performance Testing

7.3.2 Test Cases

7.3.2.1 Test Suite: Account Management

Table 7.1 Account Management Test Cases

Test ID	**Test Condition**	**Expected Output**	**Actual Output**
TC-A.1	Valid user login	User is logged in and redirected to the voter dashboard (URL: /web/app/). A session cookie is set for the user.	As Expected
TC-A.2	Invalid user login (incorrect password)	The login form (template: voting/login.html) is displayed. An error message "Invalid credentials" is displayed on the form. The session is *not* modified.	As Expected
TC-A.3	Valid admin login	Admin is logged in and redirected to the admin dashboard (URL: /admin/). A session cookie is set for the admin.	As Expected
TC-A.4	Attempt to access admin page with user	Voter is redirected to the voter dashboard (URL: /web/app/).	As Expected
TC-A.5	Attempt user login 3x with incorrect password, expect lock	The OTP should not be able to bypass or access by any means, to avoid any security threats	As Expected

TC-A.6	Successful Registration	New User to be created, the database account model to be added and configured with the correct access rights and details	As Expected

7.3.2.2 Test Suite: User Management

Table 7.2 User Management Test Cases

Test ID	Test Condition	Expected Output	Actual Output
TC-VM.1	User adds valid social media links	Links are stored, appear on public profile	As Expected
TC-VM.2	User enters invalid social media URL	Error shown, data not saved.	As Expected
TC-VM.3	User updates profile details	Changes reflected in database and frontend.	As Expected
TC-VM.4	Unauthorized user tries editing another's profile	Access denied with redirect or error	As Expected
TC-VM.5	Admin verifies submitted developer account	Status is updated, email confirmation sent	As Expected

7.3.2.3 Test Suite: QR code access & Sharing

Table 7.3 QR code management Test Cases

Test ID	Test Condition	Expected Output	Actual Output
TC-VP.1	User scans QR code	Redirected to public profile	As Expected
TC-VP.2	QR code expired or invalid	Error message displayed, access blocked	As Expected
TC-VP.3	QR access on mobile	Responsive profile page is loaded	As Expected
TC-VP.4	Unauthorized user attempts QR reuse with token change	Invalid token error shown	As Expected

7.3.2.4 Test Suite: Analytics & Paid features

Table 7.4 Analytics Test Cases

Test ID	Test Condition	Expected Output	Actual Output
TC-PDF.1	User with no analytics plan	Redirected to payment page.	As Expected
TC-PDF.2	Paid user accesses region-wise stats	Full charts loaded with filters and export options	As Expected

TC-PDF.3	Analytics load under traffic	Charts render without crash or delay.	As Expected
TC-PDF.4	Admin checks user analytics from dashboard	Correct data shown per user	As Expected

Chapter 8.0 Conclusion and Discussion

8.1 SELF-ANALYSIS OF PROJECT VIABILITIES

The ReachOut platform powered by Django on the backend side and React. js as frontend, Flutter for the mobile version, serves its purpose well as it allows a number of social media profiles to come under one umbrella in a user-friendly way and maintain one digital identity. For real-world deployment, the system is most viable for influencers, academics, and professionals with a multi-platform presence. Some notable strengths of the project are:

- Dynamic QR Code Generation: Each user receives a dynamic QR code that redirects to their ReachOut profile which makes it easy for sharing
- Clean and Scalable UI/UX: Figma, React js, and Tailwind CSS, the UI allows a truly smooth experience on web and mobile devices.
- Analytics Feature — Helps Premium users visualize analytics
- Accurate vote tallying and result visualization
- Modular architecture, making it easy to maintain and enhance

This project reflects strong understanding and practical implementation of web development concepts, database design, user flow, and software security principles.

8.2 PROBLEMS ENCOUNTERED AND POSSIBLE SOLUTIONS

Table 8.1 Problems faced

Problem	Possible Solution / Action Taken
Initial authentication using basic login was insecure	Integrated email-based Password system for secure User login
UI lacked professionalism initially	Integrated AdminLTE for a polished and modern admin interface
Email deliverability during development	Configured SMTP properly and used test email services when needed

8.3 SUMMARY OF PROJECT WORK

On The ReachOut project started to simplify the process of managing social media identity across different platforms and to make it easier for people with multiple accounts. The project was of Waterfall nature over a three-month long development cycle, during which constant iterations were made guided by mentor inputs and reviews.

Key achievements include:

- A centralized social link manager in development
- flutter mobile app (work in progress) for access on any device
- An analytics dashboard with breakdowns by region and profile insights
- OAuth integrations with major platforms (LinkedIn, Instagram, etc. — WIP)
- Comprehensive testing and validation
- Documentation and design artifacts (ER diagram, DFD, Use Case, etc.)

CHAPTER 9 LIMITATION AND FUTURE ENHANCEMENT

Limitations

ReachOut fulfills its promise of linking social profiles and providing analytics successfully, but its current iteration fails to deliver in some sections:

- Missing MFA for admin and premium users, as unauthorized access would enhance attack capabilities.
- Only web version is currently available, mobile app is still under development.
- Analytics is limited to basic visit and location-based statistics, and advanced insights such as engagement time and bounce rates are not (yet!) available.
- No real-time syncing of data from social platforms, so changes to your profile require the user to update or synchronization.
- Third-party API integrations such as Instagram or LinkedIn — subject to approval — not yet available (e.g. platform policy and documentation)
- Only English supported — no multi-language for international users.
- Lacks support for real-time data syncing across distributed servers
- No dedicated mobile app; only responsive web interface

Future Enhancements

We're training you on the data until October 2023 To enhance the Platform and potentially to offer more features, we have several updates lined up:

- Finalize Flutter, Android, and iOS components of the mobile app for all devices.
- Enable MFA for both the admin account and user accounts to enhance the platform's security.
- Link Visits: Update with Push Notifications and Emails
- Powerful Analytics including session duration, CTR and interaction hot maps
- Multi-language support to reach a wider audience.
- SMS-based OTP option in addition to email

- Blockchain-based vote integrity and tamper-proof logging
- AI-powered analytics for result predictions or voting pattern insights
- REST API integration to support mobile applications