

DIGITAL INDIA SMART SUBSIDY SYSTEM

A PROJECT REPORT

Submitted by,

Mr. ANSHU KUMAR 20201CSE0438

Mr. SREEJA GVS 20201CSE0775

Under the guidance of,

Dr. Thrimoorthy N

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At



PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report “**Digital India Smart Subsidy System**” being submitted by “Anshu kumar , Sreeja GVS” bearing roll number(s) “20201CSE0438, 20201CSE0775,” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. Thrimoorthy N

Assistant Professor

School of CSE

Presidency University

Dr. PALLAVI

HoD

School of CSE

Presidency University

Dr. C. KALAIARASAN

Associate Dean

School of CSE

Presidency University

Dr. L. SHAKKEERA

Associate Dean

School of CSE

Presidency University

Dr. SAMEERUDDIN KHAN

Dean

School of CSE

Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled Digital india smart subsidy system in partial fulfilment for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a record of our own investigations carried under the guidance of Dr. Thrimoorthy N, Assistant Professor , School of Computer Science and Engineering, Presidency University, Bengaluru.

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NO	SIGNATURE
ANSHU KUMAR	20201CSE0438	
SREEJA GVS	20201CSE0775	

ABSTRACT

The Digital India Smart Subsidy System Web Portal represents a pivotal component within the broader initiative to modernize and optimize subsidy distribution across diverse sectors. This user-friendly online platform serves as the interface linking beneficiaries, government authorities, and administrators, streamlining subsidy-related processes. With a focus on accessibility, transparency, and efficiency, the portal incorporates cutting-edge web technologies. Beneficiaries can easily register, manage profiles, and apply for subsidies, with real-time tracking functionalities to monitor application status. The system ensures secure user authentication, electronic document submission, and verification processes. Government officials benefit from a dedicated dashboard to oversee subsidy applications, allocations, and disbursements. Additionally, the portal integrates payment gateways for secure fund transfers, a notification system for real-time updates, and a data analytics dashboard to inform subsidy allocation decisions. The inclusion of multilingual support and a robust helpdesk fosters inclusivity, while training resources empower users to navigate the portal effectively. Ultimately, the Digital India Smart Subsidy System Web Portal plays a transformative role in advancing subsidy distribution through its emphasis on user empowerment, governmental efficiency, and digital innovation.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science and Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. C. Kalaiarasan and Dr. Shakkeera L**, School of Computer Science and Engineering, Presidency University and **Dr. Pallavi R**, Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr. Thrimoorthy N, Asst-Prof**, School of Computer Science and Engineering, Presidency University for her inspirational guidance, valuable suggestions and providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators **Mr. Zia Ur Rahman and Mr. Peniel John Whistely**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

ANSHU KUMAR

SREEJA G V S

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 9.1	Gantt Chart	15
2	Table 9.2	Time Estimation	16

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 8.1	Sequence Diagram	13
2	Figure 8.2	Use case diagram	13
3	Figure 8.3	Activity diagram	14
5	Figure B.1	Registration page	31
6	Figure B.2	Login page	31
7	Figure B.3	List of Subsidy	32
8	Figure B.4	Application Form	32
9	Figure B.5	Application Details	33
10	Figure B.6	Home Page	33

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	ACKNOWLEDGMENT	v
	LIST OF TABLES	vi
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
	1.1 Background	1
	1.2 Key Features and Functionality	2
	1.2.1 Conversational Interface	2
	1.2.2 Order Status and History	2
	1.2.3 Backend Integration with Supabase	2
	1.3 Technological Underpinnings	2
	1.3.1 Django framework	2
	1.4 User Interaction Flow	2
	1.4.1 Initialization	3
	1.4.2 Order-related Inquiries	3
2.	LITERATURE SURVEY	4
3.	RESEARCH GAPS OF EXISTING METHODS	5
	3.1 Effectiveness and Impact Assessment	5
	3.2 User Behavior and Adoption	5
	3.3 Technological Challenges	5
4.	PROPOSED METHODOLOGY	6
	4.1 User-Centric Design	6
	4.2 Collaborative Governance	6
	4.3 Agile Development Approach	6

	4.4 Data Security Measures	6
	4.5 Public Awareness Campaign	6
5.	OBJECTIVES	7
	5.1 Efficient and Transparent Distribution	7
	5.2 Inclusion and Accessibility	7
	5.3 Data-Driven Decision Making	7
	5.4 Secure and Scalable Infrastructure	7
	5.5 Enhanced User Experience	7
6.	SYSTEM DESIGN AND IMPLEMENTATION	8
	6.1 Hardware System Requirement	8
	6.1.1 Hardware Requirements	8
	6.1.2 Software Requirements	8
	6.2 Design and Development	8
	6.2.1 Frontend Development	8
	6.2.2 Backend Integration	8
	6.2.3 Database Integration	9
7	TESTING	10
	7.1 Introduction	10
	7.2 Types of Testing	10
	7.2.1 Specification Testing	10
	7.2.2 Module Level Testing	10
	7.2.3 Unit Testing	10
	7.2.4 Integration Testing	10
	7.2.5 Validation Testing	10
	7.2.6 Recovery Testing	11
	7.2.7 Security Testing	11
	7.2.8 Performance Testing	11
	7.2.9 Blackbox Testing	11
	7.2.10 Output Testing	11

	7.2.11 User Acceptance Testing	12
8	DIAGRAMS	13
	8.1 Sequence Diagram	13
	8.2 Use Case Diagram	13
	8.3 Activity Diagram	14
9	TIMELINE FOR EXECUTION OF PROJECT	15
10	OUTCOMES	18
	10.1 Order Tracking Chatbot	18
	10.1.1 User Authentication	18
	10.1.2 Home Page and Chat Interface	18
	10.1.3 Options Selection and Functionality	18
	10.1.4 Order Tracking and Status Updates	18
	10.1.5 Database Integration with Supabase (PostgreSQL)	18
	10.1.6 User Management and Security	18
	10.1.7 Error Handling and Assistance	18
	10.1.8 Scalability and Future Enhancements	19
	10.1.9 Launch and Marketing	19
11	RESULT AND DISCUSSIONS	20
	11.1 Smart Subsidy System Using Frame Work Django	20
	11.1.1 User Authentication	20
	11.1.2 Chatbot Functionality	20
	11.1.3 Database Integration	20
	11.1.4 Cross-Platform Compatibility	20
	11.1.5 Responsive User Interface	20
	11.1.6 Order Tracking and Personalization	21
	11.1.7 Security Measures	21
	11.1.8 Documentation	21
	11.1.9 Future Enhancements	21

	11.2 Cement Chatbot Application using Flutter	21
	11.2.1 Positive User Experience	21
	11.2.2 Efficiency in Customer Service	22
	11.2.3 Cross-Platform Accessibility	22
	11.2.4 Secure and Confidential	22
	11.2.5 Foundation for Future Growth	22
12	CONCLUSION	23
	APPENDIX-A: PSEUDOCODE	24
	APPENDIX-B: SCREENSHOTS	30
	APPENDIX-C: ENCLOSURES	33
	PLAGIARISM REPORT	35
	SUSTAINABLE DEVELOPMENT GOALS	36

CHAPTER-1

INTRODUCTION

The introduction of the Digital India Smart Subsidy System marks a significant milestone in India's ambitious journey towards a digitally empowered society. In the context of subsidy distribution, a critical aspect of socio-economic governance, India has grappled with persistent challenges related to fairness, efficiency, and transparency. Traditional methods of subsidy disbursal have been marred by systemic delays, leakages, and a lack of visibility, thwarting the intended positive impact on the diverse and extensive population. Recognizing the imperative for a transformative shift, the Indian government has launched the Digital India Smart Subsidy System. This groundbreaking initiative is positioned as a beacon of innovation and efficiency, strategically designed to leverage the immense potential of digital technologies. The multifaceted challenges, deeply entrenched in the country's socio-economic fabric, necessitate a holistic approach to subsidy management. It is within this context that the Digital India Smart Subsidy System emerges as a comprehensive solution, integrating cutting-edge digital platforms, advanced data analytics, and secure transaction mechanisms. At its core, the rationale behind this initiative is rooted in addressing long standing issues that have impeded the equitable and transparent distribution of subsidies. Inefficiencies, delays, and leakages have plagued traditional subsidy systems, thwarting the optimal realization of socio-economic benefits. The Digital India Smart Subsidy System is a strategic response to these challenges, driven by the vision to eliminate inefficiencies, reduce leakages, and ensure that subsidies reach their intended beneficiaries promptly and transparently. The project's focus extends beyond mere technological integration; it aspires to create a seamless, accountable, and inclusive subsidy ecosystem. The digital transformation is not viewed in isolation but is intricately linked to broader national goals, particularly those related to financial inclusion and socio-economic development. By embracing digital technologies, the government aims not only to modernize subsidy distribution but also to catalyze a positive shift in the socio-economic landscape, ensuring that the benefits of subsidies are maximized for the diverse population of India. In essence, the Digital India Smart Subsidy System heralds a new era in subsidy management, propelled by a vision of efficiency, transparency, and inclusive development. As the initiative unfolds, it is poised to redefine the subsidy distribution paradigm, setting the stage for a more digitally advanced, accountable, and accessible governance framework in India.

1.1 Background

The foundation of the smart subsidy project lies in the convergence of blockchain technology, and user interaction. The project utilizes Django, a popular open-source UI software development toolkit, to build a cross-platform software application. This ensures

that users on OS and iOS platforms can access the features and capabilities offered by the subsidy.

1.2 Key Features and Functionality

The SMART SUBSIDY application introduces several key features aimed at enhancing user experience and providing valuable functionalities.

1.2.1 Conversational Interface

The heart of the application lies in its conversational interface, allowing users to directly apply for subsidy without and middle man.

1.2.2 Order Status and History

Users can effortlessly check the status of their Applied application , review order history, and obtain real-time updates on their deliveries.

1.2.3 Backend Integration with Supabase

The Supabase backend serves as a robust foundation for managing user data, storing order details, and maintaining a record of conversations. This integration ensures data consistency, security, and scalability, creating a seamless experience for users.

1.3 Technological Underpinnings

The project employs a technology stack that synergizes various tools and frameworks to achieve its objectives/

1.3.1 Django framework

Utilizing the Django framework enables the development of a cross-platform with a single codebase. This ensures consistency in user experience across different devices.

1.4 User Interaction Flow

The smart subsidy application follows an intuitive user interaction flow, ensuring a seamless experience for users.

1.4.1 Initialization

Upon launching the software, users are greeted with an initial popup window for registration.

1.4.2 Order-related Inquiries

For tasks like checking order status or reviewing order history, the software communicates with the Supabase backend to fetch relevant information. The responses are then presented to users in a conversational format.

CHAPTER-2

LITERATURE SURVEY

Numerous scholarly articles, research papers, and case studies have emphasized the significance of employing digital platforms for subsidy distribution. These studies often highlight the following points: Efficiency: Digital systems can significantly improve the efficiency of subsidy distribution by reducing bureaucratic hurdles and manual processes. Transparency: They enable transparent tracking and monitoring of subsidy transactions, ensuring accountability in the system. Financial Inclusion: Digital platforms can help include marginalized sections of society, ensuring they receive entitled subsidies without intermediaries. Technological Integration: Integrating various technologies like biometrics, Aadhar card, mobile applications, and web portals can enhance the system's effectiveness.

Advantages:

1. **Transparency and Accountability:** Digital platforms can offer a transparent trail of subsidy disbursements, ensuring accountability and minimizing corruption.
2. **Efficiency:** Automation and digitization reduce paperwork, processing time, and human errors, enhancing operational efficiency.
3. **Financial Inclusion:** Digitization allows easier access to subsidies for remote and marginalized communities.
4. **Data Analytics:** Data collected through digital systems can be utilized for analytics, enabling better policymaking and targeted subsidy distribution.
5. **Convenience:** Beneficiaries can access and manage their subsidy information conveniently through web or mobile interfaces.

Disadvantages:

1. **Digital Divide:** Some segments of the population might face challenges in accessing or using digital platforms due to technological literacy or lack of resources.
2. **Security Concerns:** Data breaches, cyber threats, and privacy issues could compromise sensitive beneficiary information if adequate security measures are not implemented.
3. **Infrastructure Challenges:** Inadequate internet connectivity or power supply in certain regions might hinder the seamless functioning of digital platforms.
4. **Dependency on Technology:** System failures, technical glitches, or server downtime could disrupt subsidy distribution.
5. **Exclusion Errors:** Errors in data entry or authentication mechanisms might lead to the exclusion of eligible beneficiaries or inclusion of ineligible ones.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

Research gaps in the field of smart subsidies can vary depending on the specific focus and context of the study. However, here are some general research gaps that researchers may consider exploring in the realm of smart subsidies:

3.1 Effectiveness and Impact Assessment

- Evaluate the effectiveness of smart subsidy programs in achieving their intended goals and objectives.
- Assess the impact of smart subsidies on economic, social, and environmental outcomes.
- Investigate the long-term sustainability and scalability of smart subsidy initiatives.

3.2 User Behavior and Adoption

- Understand the factors influencing user acceptance and adoption of smart subsidy technologies or platforms.
- Explore the behavioral aspects of subsidy recipients and how they interact with smart subsidy systems.
- Investigate the role of education and awareness in promoting the adoption of smart subsidy programs.

3.3 Technological Challenges

- Examine the technological challenges associated with implementing smart subsidy solutions, such as data security, privacy concerns, and interoperability issues.
- Investigate the potential of emerging technologies (e.g., blockchain, artificial intelligence)

CHAPTER-4

PROPOSED METHODOLOGY

The methodology for implementing the Digital India Smart Subsidy System involves a systematic approach encompassing needs assessment, feasibility study, system design, technology stack selection, development, testing, pilot implementation, full-scale rollout, monitoring, evaluation, capacity building, and continuous improvement. This comprehensive process ensures the seamless integration of digital technologies, user-friendly interfaces, and robust security measures to revolutionize subsidy distribution in India, promoting efficiency, transparency, and inclusivity.

4.1 User-Centric Design:

Prioritize the needs and preferences of end-users in the development process, ensuring a seamless and intuitive experience for both beneficiaries and administrators.

4.2 Collaborative Governance

Foster collaboration between government agencies, technology experts, and stakeholders to create a unified and interoperable system, promoting efficiency and reducing redundancy.

4.3 Agile Development Approach:

Adopt an agile methodology to facilitate iterative development, allowing for continuous feedback, adjustments, and improvements throughout the project lifecycle.

4.4 Data Security Measures

Implement robust data security protocols and encryption mechanisms to safeguard sensitive information, ensuring the privacy and integrity of beneficiary data.

4.5 Public Awareness Campaign:

Launch a targeted public awareness campaign to educate beneficiaries about the advantages of the Digital India Smart Subsidy System, promoting increased adoption and trust in the digital subsidy distribution platform.

CHAPTER-5

OBJECTIVES

Establish a system where every activity will be monitored via blockchain.

Application Framework: Choose a web application framework, such as Django for building the portal's backend logic and APIs. This infrastructure will facilitate seamless interactions, allowing the software to send and receive data securely. The architecture aims for modularity and flexibility, supporting the evolution of the system as requirements change.

5.1 Efficient and Transparent Distribution

The system seeks to streamline subsidy distribution processes, minimizing delays and enhancing transparency to build trust among beneficiaries and stakeholders.

5.2 Inclusion and Accessibility

A core objective is to make subsidies accessible to all eligible beneficiaries, including those in remote areas.

5.3 Data-Driven Decision Making

Through the implementation of data analytics tools, the project aims to facilitate informed decision-making by analyzing demographic, economic, and social data. This approach ensures targeted subsidy delivery based on specific needs and indicators.

5.4 Secure and Scalable Infrastructure

The project places a strong emphasis on the creation of a secure and scalable digital infrastructure capable of handling the vast amounts of data associated with subsidy distribution.

5.5 Enhanced User Experience

By developing a user-friendly web portal the system aims to enhance the overall user experience for both beneficiaries and administrators.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 Hardware System Requirement

We need both hardware and software components to ensure the portal's functionality, security, and scalability

6.1.1 Hardware Requirements

- PC or Laptop
- Operating System: a secure and reliable operating system for our servers.

6.1.2 Software Requirements:

- Frontend Development: Implement HTML, CSS, and JavaScript for the portal's user interface and frontend interactions.
- Programming Languages: Using programming language Python to develop the portal's software components.
- Application Framework: Choose a web application framework, such as Django for building the portal's backend logic and APIs.
- Database Management System (DBMS): DBMS like MySQL to manage data, and other information.

6.2 Design and Development

6.2.1 Frontend Development

We used HTML/CSS, JAVASCRIPT for the frontend to interact with the customers responsively.

Responsive Design: Implement a responsive design to ensure the application adapts well to various screen sizes and orientations.

6.2.2 Backend Integration

Developing the server-side application involves creating the logic to handle the customer issues and interactions and database communication. Key steps in the implementation include:

API Endpoints: Define Django API endpoints that the frontend can interact with. These endpoints handle requests related to user authentication, and database interactions.

Server Logic: Implement the server-side logic for processing user inputs, managing software responses, and coordinating interactions with the database.

User Authentication: Develop mechanisms for user authentication, including login and registration functionalities. Ensure secure communication between the frontend and backend.

6.2.3 Database Integration

Integrating a database system involves storing and retrieving data related to product information and user details. The implementation may include:

Database Schema: Design a database schema that accommodates tables for storing product details, user information, and any other relevant data.

Database Connectivity: Establish connections between the backend server and the database, ensuring secure and efficient data retrieval and storage.

Query Execution: Implement SQL queries to perform CRUD (Create, Read, Update, Delete) operations on the database. This includes fetching product details, storing user information, and managing order history.

Data Security: Implement security measures such as encryption and access controls to protect sensitive user data stored in the database.

CHAPTER-7

TESTING

7.1 Introduction

Following the completion of the development phase for any computer-based system, the subsequent intricate and time-consuming stage involves system testing. It is during this testing period that the development company gains insight into the extent to which user requirements have been fulfilled and other relevant details.

7.2 Types Of Testing

7.2.1 Specification Testing:

We can establish the expected behavior and performance criteria for the program under different conditions. This testing involves a comparative analysis of the system's performance against its specified requirements.

7.2.2 Module Level Testing:

Within this approach, errors are identified within each individual module, fostering an environment where programmers can locate and address errors without impacting other modules.

7.2.3 Unit Testing

Unit testing is centered on validating the functionality of the smallest software module. This involves scrutinizing the local data structure to ensure the temporary storage of data maintains its integrity throughout all stages of the algorithm's execution. Additionally, boundary conditions are tested to verify that the module functions correctly at established limits or restrictions, ensuring robust performance.

7.2.4 Integration Testing:

Data can undergo testing across an interface, where one module might unintentionally impact another adversely. Integration testing is a systematic approach to building a program structure while simultaneously conducting tests to reveal errors related to interfacing between modules.

7.2.5 Validation Testing:

It commences after the successful assembly of integration testing. Validation is deemed

successful when the software operates in a manner reasonably acceptable to the client. The primary focus of validation lies in the data entry operation, where the highest likelihood of entering incorrect data exists. Additional validation processes occur throughout all stages, ensuring accurate details and data entry to obtain the desired results.

7.2.6 Recovery Testing:

Recovery Testing is a system that deliberately induces failures in the software, examining whether the recovery process is executed correctly. If recovery is automated, both re-initialization and data recovery are assessed for accuracy.

7.2.7

S

Security Testing:

Security testing aims to confirm that the protective mechanisms integrated into the system can effectively shield it from unauthorized penetration. The tester might try obtaining passwords through external administrative methods, deploy custom-designed software to dismantle any defenses, and intentionally introduce errors to assess the system's resilience.

7.2.8 Performance Testing:

Performance testing is employed to evaluate the runtime performance of software within the framework of an integrated system. These tests are frequently linked with stress testing and necessitate the incorporation of software instrumentation.

7.2.9 Blackbox Testing:

Black-box testing concentrates on the functional requirements of software, facilitating the derivation of sets of input conditions that thoroughly exercise all functional aspects of a program. The primary aim of black-box testing is to identify errors in the following categories:

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structures or external database access
4. Performance errors.

7.2.10 Output Testing:

Following the completion of validation testing, the subsequent stage involves output testing of the proposed system. A system is not deemed useful until it produces the required output in the specified format. Output format is assessed in two aspects: the screen format and the printer format.

7.2.11 User Acceptance Testing

User Acceptance Testing is a pivotal element in the success of any system. The system in question is evaluated for user acceptance by maintaining constant communication with potential system users during development and making necessary adjustments as needed.

CHAPTER-8

DIAGRAMS

8.1 Sequence Diagram

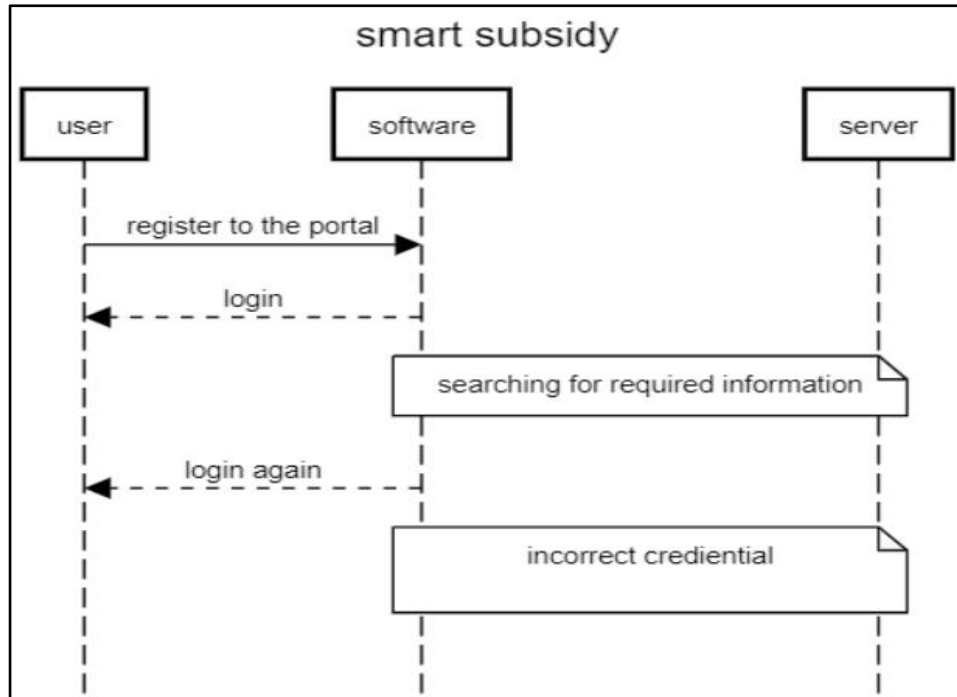


Figure 8.1 Sequence Diagram

8.2 Use-case Diagram

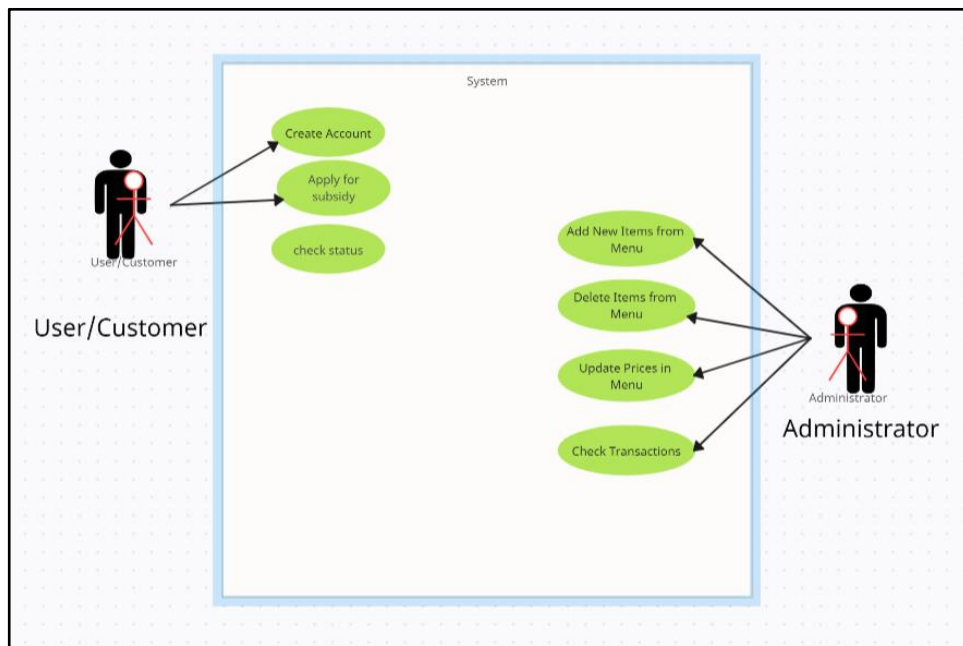


Figure 8.2 Use Case Diagram

8.3 Activity diagram

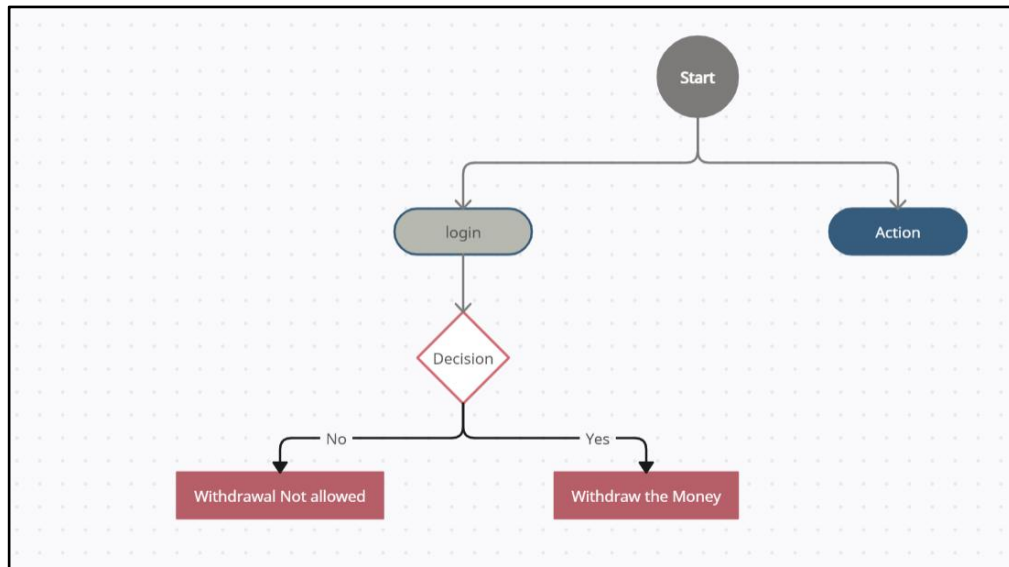


Figure 8.3 Activity Diagram

CHAPTER-9

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

TASK	DURATION (Weeks)	START DATE	END DATE	DEPENDENCIES
Project Initiation	1	08/10/2023	15/08/2023	-
Requirement Analysis	1	17/08/2023	24/10/2023	Project Initiation
System Design (Database, User Interface (UI) & Integration)	2	25/10/2023	07/11/2023	Requirement Analysis
Development (Frontend & Backend)	4	08/11/2023	29/11/2023	System Design (All)
Database Implementation	1	30/11/2023	07/12/2023	Development (Backend)
Testing (Unit & Integration) + User Acceptance Testing	2	08/12/2023	22/12/2023	Testing (All)
Project Documentation & Closure	1	23/12/2023	30/12/2023	Monitoring and Maintenance Planning

Table 9.1: Gantt Chart

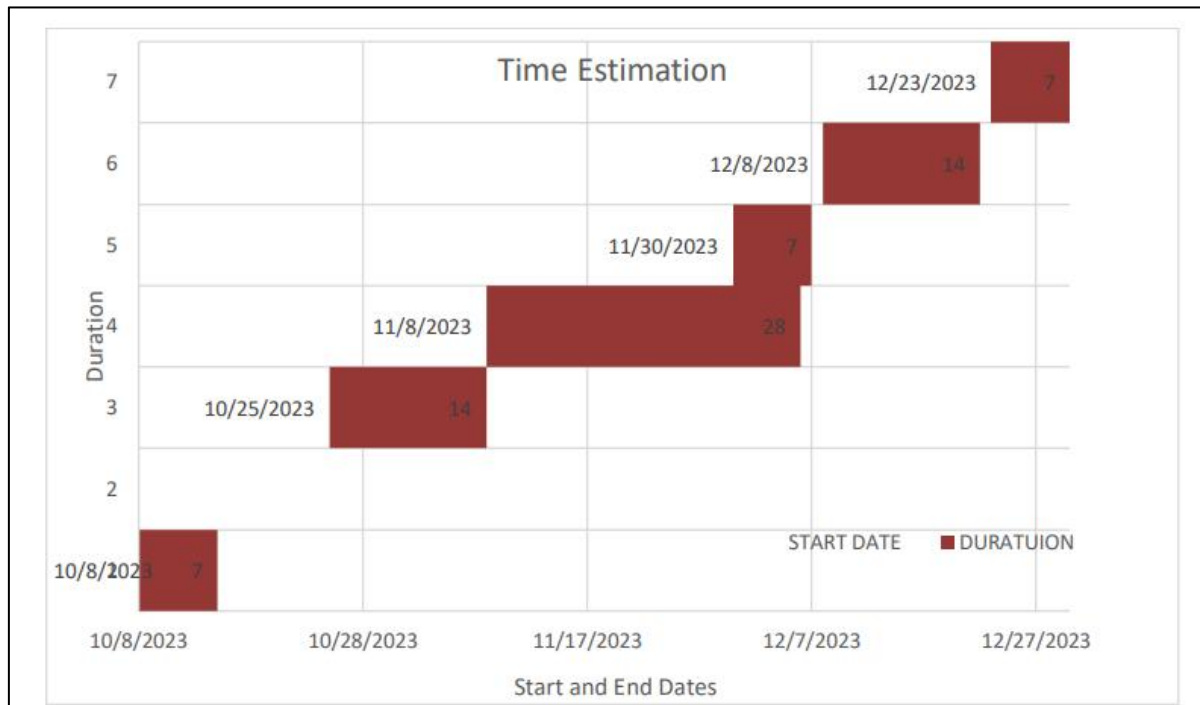


Table 9.2 Time Estimation

The Gantt chart provides an overview of the project's organisational structure. Its left side lists tasks, most likely divided into phases or significant checkpoints. Unfortunately, fuzzy wording makes individual task names harder to read. The project will take place over roughly 11 weeks, starting on October 8, 2023, and ending on December 27, 2023. Horizontal bars are used to visually express task durations; the length of each bar indicates the task's anticipated duration. For instance, the first assignment has a 3-week duration planned, with a timeline that runs from October 8 to October 28. Although there are no obvious arrows or lines that connect jobs, suggesting possible independence, it is possible that dependencies exist but are not shown in this version. Unfortunately, the chart is devoid of progress indicators, like marks or shading, which could offer valuable information on task completion. To sum up, although the Gantt chart offers a basic grasp of the project's schedule and task durations, the omitted task specifics, unclear dependencies, and lack of progress tracking components all make it difficult to fully understand the project plan. Improving these elements would greatly aid in creating a visualisation of the project's trajectory that is both more instructive and useful.

- **Task Listing:** The project's tasks are listed on the left side of the Gantt chart, most likely divided into phases or milestones. Unfortunately, fuzzy wording makes it difficult to understand precise task names.
- **Chronology:** The project's chronology is shown in the upper section of the chart. It runs from October 8, 2023, to December 27, 2023, or roughly 11 weeks.
- **Task Durations:** Every task is shown as a horizontal bar across the timeline, and the length of the bar indicates how long the work is expected to take. For example, the first assignment has a scheduled period of three weeks, spanning from October 8 to October 28.
- **Dependencies:** Potential independence is shown by the lack of obvious arrows or lines connecting tasks. It is possible, therefore, that dependencies exist even though they aren't stated clearly in this chart.
- **Progress tracking:** There are no markings or shading on the chart to indicate the status of the project. These components are necessary to comprehend the extent of each task's completion.

In summary, the Gantt chart provides a basic overview of the project's timetable and task durations, but it is not very useful for tracking progress, providing precise task information, or defining dependencies. By providing a more thorough and useful depiction of the project plan, these improvements would greatly increase the chart's usefulness.

CHAPTER-10

OUTCOMES

10.1 Order Tracking Chatbot

10.1.1 User Authentication:

- Login Page: Existing users can securely log in using their credentials, ensuring data privacy and restricted access to authorized users.
- Register Page: New users can create accounts by providing necessary details, enabling a seamless onboarding process.

10.1.2 Home Page and Chat Interface

- User-Friendly Interface: The home page serves as the central hub for users to navigate through the subsidy functionalities.

10.1.3 Options Selection and Functionality

- Menu Options: Users can select from various menu options presented by the software, such as tracking an order, checking status, or requesting assistance.

10.1.4 Order Tracking and Status Updates

- Order Tracking: Users can inquire about the status and location of their orders through the software.

10.1.5 Database Integration with Supabase (PostgreSQL)

- Backend Database: Order details, user information, and other relevant data are securely stored in the backend database using MySQL, integrated with Django.

10.1.6 User Management and Security

- User Profiles: Existing users have personalized profiles with order history, preferences, and account details.
- Secure Transactions: The software ensures secure authentication and transactions, protecting user data and maintaining confidentiality.

10.1.7 Error Handling and Assistance

- User Guidance: The software offers assistance and guidance in case of user errors or misunderstandings, ensuring a smooth and user-friendly experience.
- Error Logging: Any errors or issues encountered during user interactions are logged for analysis and improvement.

10.1.8 Scalability and Future Enhancements

- Scalable Architecture: The project is designed with scalability in mind, allowing for the integration of additional features and accommodating a growing user base.
- Feedback Mechanism: Implement a feedback mechanism to gather user opinions and suggestions for continuous improvement.

10.1.9 Launch and Marketing

- Launch Strategy: Plan and execute a strategic launch of the order tracking software, including marketing initiatives to promote user adoption.

CHAPTER-11

RESULTS AND DISCUSSIONS

11.1 Smart Subsidy System Using Frame Work Django

11.1.1 User Authentication

- Result: Successful implementation of a secure user authentication system with both login and registration functionalities.
- Impact: Users can securely log in to access personalized features and register as new users to create accounts.

11.1.2 Chatbot Functionality

- Result: Integration of an NLP-based chatbot that effectively understands and responds to user queries.
- Impact: Users can interact with the chatbot in a conversational manner, making inquiries about product details, pricing, and placing orders.

11.1.3 Database Integration

- Result: Successful integration with Supabase, utilizing PostgreSQL for efficient storage and retrieval of product information and user details.
- Impact: The application seamlessly interacts with the backend database, ensuring accurate and up-to-date information for users.

11.1.4 Cross-Platform Compatibility

- Result: Implementation using Flutter ensures a consistent and seamless experience on both Android and iOS devices.
- Impact: The application is accessible to a broader user base, catering to users with diverse mobile platforms.

11.1.5 Responsive User Interface

- Result: Utilization of Flutter widgets for frontend development ensures an intuitive and responsive user interface.
- Impact: Users experience a visually appealing and easy-to-navigate interface, enhancing overall satisfaction.

11.1.6 Order Tracking and Personalization

- Result: Successful implementation of features such as order tracking, user preferences, and personalized user profiles.
- Impact: Users can track their orders, save preferences, and benefit from a customized experience, contributing to improved customer satisfaction.

11.1.7 Security Measures

- Result: Implementation of security measures, including encryption and access controls, to protect user data.
- Impact: User data is safeguarded, ensuring confidentiality and building trust among users.

11.1.8 Documentation

- Result: Comprehensive documentation covering the project architecture, codebase, and database schema.
- Impact: A valuable resource for future maintenance, knowledge sharing, and onboarding new developers.

11.1.9 Future Enhancements

- Result: The project lays a foundation for future enhancements, including integration with customer support, voice recognition, and CRM systems.
- Impact: The application is positioned for continuous improvement and adaptation to emerging technologies and customer needs.

11.2 Cement Chatbot Application using Flutter

The Cement Chatbot Application using Flutter has achieved its objectives by successfully implementing a robust and user-friendly platform for customer interaction and assistance within the cement industry. The conclusion of the project can be summarized as follows:

11.2.1 Positive User Experience

- The application provides a positive and seamless user experience, allowing customers to interact with the chatbot naturally and access information effortlessly.

11.2.2 Efficiency in Customer Service

- The integration of a chatbot streamlines customer service processes, allowing users to inquire about products, place orders, and track deliveries efficiently.

11.2.3 Cross-Platform Accessibility

- The choice of Django ensures cross-platform compatibility, making the software accessible to a wide range of users across OS and iOS devices.

11.2.4 Secure and Confidential

- Security measures implemented in the application ensure the confidentiality of user data, contributing to a secure and trustworthy environment.

11.2.5 Foundation for Future Growth

- The project sets the stage for future enhancements, demonstrating adaptability to emerging technologies and customer service trends.

CHAPTER-12

CONCLUSION

The Digital India Smart Subsidy System converge towards a transformative shift in subsidy distribution, aligning with the government's broader vision of a digitally inclusive and transparent governance. The streamlined processes, reduced processing time, and real-time monitoring mechanisms promise tangible efficiency gains, ensuring subsidies reach beneficiaries promptly and with unprecedented transparency. The infusion of data analytics, user-centric design, and digital payment solutions not only enhance the user experience but also contribute to financial inclusion and evidence-based decision-making. Moreover, the system's commitment to security and scalability signifies a robust foundation for long-term adaptability, fostering public trust and setting a precedent for effective digital governance practices in India. As this initiative unfolds, its multifaceted impact positions it as a catalyst for positive societal change, optimizing subsidy distribution and contributing to India's trajectory as a digitally empowered and economically vibrant nation. The Digital India Smart Subsidy System represents a pivotal milestone in the evolution of subsidy management, promising a holistic transformation in operational efficiency, financial inclusion, and governance practices. The expected outcomes, ranging from enhanced transparency and data-driven decision-making to improved user experiences, collectively reinforce the potential of this initiative to redefine the socio-economic landscape. As the system takes root and evolves, it not only addresses the immediate challenges of subsidy distribution but also establishes a resilient framework for future advancements in digital governance, marking a significant stride towards a more inclusive and digitally empowered India.

APPENDIX-A

PSUEDOCODE

1. Model.py

```
from django.db import models
```

```
class Contact(models.Model):
```

```
    name=models.CharField(max_length=20)
```

```
    email=models.CharField(max_length=20)
```

```
    message=models.TextField()
```

```
class Schemesinfo(models.Model):
```

```
    schemename =models.CharField(max_length=50)
```

```
    citizenship =models.CharField(max_length=50)
```

```
    State =models.CharField(max_length=50)
```

```
    age =models.CharField(max_length=50)
```

```
    gender =models.CharField(max_length=50)
```

```
    caste =models.CharField(max_length=50)
```

```
    rationcard =models.CharField(max_length=50)
```

```
    employmentStatus =models.CharField(max_length=50)
```

```
    aadhaarnumber =models.CharField(max_length=12)
```

```
    link=models.URLField()
```

```
class Applications(models.Model):
```

```
    first_name = models.CharField(max_length=50)
```

```
    lastname = models.CharField(max_length=50)
```

```
    date_of_birth = models.DateField()
```

```
    citizenship = models.CharField(max_length=50)
```

```
    state = models.CharField(max_length=50)
```

```
    age = models.CharField(max_length=50)
```

```
    gender = models.CharField(max_length=10)
```

```
    caste = models.CharField(max_length=20)
```

```
    rationcard = models.CharField(max_length=20)
```

```
    employmentStatus = models.CharField(max_length=50)
```

```
aadhaarnumber = models.CharField(max_length=50)
schemename = models.CharField(max_length=50)
is_approved = models.BooleanField(default=False)
# Create your models here
```

2. Admin.py

```
from django.contrib import admin
from service.models import Contact, Schemesinfo, Applications
from django.contrib.auth.admin import UserAdmin

class ContactAdmin(admin.ModelAdmin):
    list_display=('name', 'email', 'message')
admin.site.register(Contact, ContactAdmin)

class SchemesinfoAdmin(admin.ModelAdmin):
    list_display=('schemename', 'citizensip', 'State', 'age', 'gender', 'caste', 'rationcard',
'employmentStatus', 'aadhaarnumber', 'link')
admin.site.register(Schemesinfo, SchemesinfoAdmin)

# class ApplicationsAdmin(admin.ModelAdmin):
#     list_display=('first_name', 'lastname', 'date_of_birth', 'citizensip', 'state',
#         'age', 'gender', 'caste', 'rationcard', 'employmentStatus', 'aadhaarnumber', 'schemename')
# admin.site.register(Applications, ApplicationsAdmin)

def approve_applications(modeladmin, request, queryset):
    queryset.update(is_approved=True)

approve_applications.short_description = "Approve selected applications"

class ApplicationsAdmin(admin.ModelAdmin):
    list_display = ['first_name', 'lastname', 'date_of_birth', 'citizensip', 'state',
        'age', 'gender', 'caste', 'rationcard', 'employmentStatus', 'aadhaarnumber', 'schemename',
'is_approved']
```

```
def save_model(self, request, obj, form, change):
    # Check if the is_approved checkbox is checked in the admin form
    is_approved = form.cleaned_data.get('is_approved')

    if is_approved:
        obj.is_approved = True
        obj.save()
    else:
        # If not approved, you might want to take some other action or not save at all
        # For example, you can redirect the user to a rejection page or raise an exception
        pass

admin.site.register(Applications, ApplicationsAdmin)

# Register your models here.
```

3. Application.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Patient Information Page</title>
    {% load static %}
    <link rel="stylesheet" href="{% static 'css/Applicationform.css' %}">
</head>
<body>
    <header class="header">
        <div class="innerheader">
            <div class="logo">
                <!--  -->
            </div>
            <H3> Digital India Smart Subsidy System </H3>
            <nav>
```

```
<ul>

    <li><a href="{% url 'homepage' %}">Home</a></li>
    <li><a href="{% url 'Dash' %}">Schemes</a></li>
    <li><a href="{% url 'aboutus' %}">About Us</a></li>
    <li><a href="{% url 'services' %}">Services</a> </li>
    <li><a href="{% url 'contactus' %}">Contact Us</a> </li>
    <li><a href="{% url 'logout_user' %}">Logout User</a> </li>

    <!-- <li><button type="submit">Register </button> </li> -->
</ul>
</nav>
<nav>
    <ul>
        <li> <a href="{% static 'css/Report.pdf' %}" download="Report.pdf"><button
type="">Download Report </button> </a> </li>
    </ul>
</nav>
</div>
</header>
```

```
<h4 class="note">Note: Please fill all the information as accurately as possible.
The information you provide will assist in formulating a complete health profile.
All Answers are confidential.</h4>
```

```
<div class="mainwrap">
    <div class="wrapp">
        
    </div>
    <div class="wrapper">
        <div class="title">
            Application Form
        </div>
        <form action="" method="POST">
```

```
{% csrf_token %}
<div class="form">
  <div class="inputfield" >
    <label>First Name</label>
    <input type="text" name="first_name" class="input" required>
  </div>
  <div class="inputfield" >
    <label>Last Name</label>
    <input type="text" name="lastname" class="input">
  </div>
  <div class="inputfield" name="date_of_birth">
    <label>Date Of Birth</label>
    <input type="" name="date_of_birth" class="input" placeholder="YYYY-MM-
DD"required>
  </div>
  <div class="inputfield" >
    <label>citizensip</label>
    <input type="text" name="citizensip" class="input" required>
  </div>
  <div class="inputfield" >
    <label>State</label>
    <input type="text" name="state" class="input" required>
  </div>
  <div class="inputfield" >
    <label>Age</label>
    <input type="number" name="age" class="input" min="0" required>
  </div>
  <div class="inputfield" >
    <label>Gender</label>
    <input type="text" name="gender" class="input" required>
  </div>
  <div class="inputfield" >
    <label>Caste</label>
    <input type="text" name="caste" class="input" required>
```

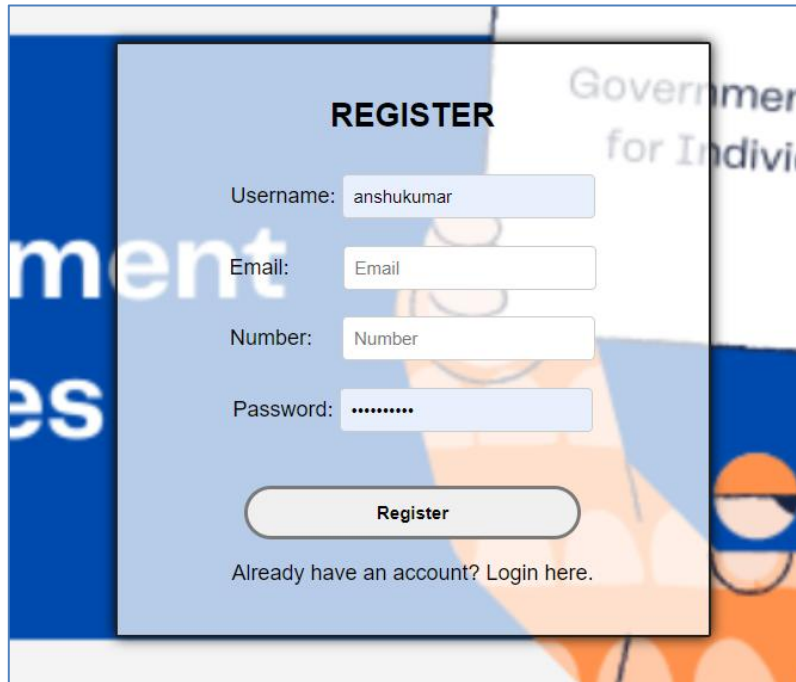
```
</div>
<div class="inputfield" >
  <label>RationCard No</label>
  <input type="text" name="rationcard" class="input" maxlength="12" required>
</div>
<div class="inputfield" >
  <label>Employment Status</label>
  <input type="text" name="employmentStatus" class="input" required>
</div>
<div class="inputfield" >
  <label>Aadhaar Number</label>
  <input type="text" name="aadhaarnumber" class="input" required>
</div>
<div class="inputfield" >
  <label>Scheme Name</label>
  <input type="text" name="schemename" class="input" required>
</div>

  <div class="inputfield">
    <button class="btn">Submit</button>
  </div>

</div>
</form>
</div>
</div>
</body>
</html>
```

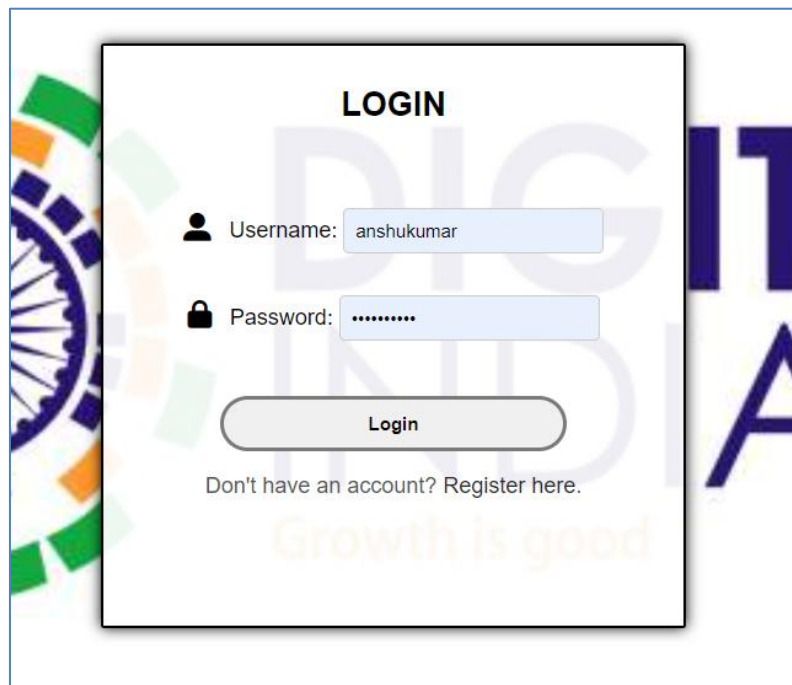

APPENDIX-B

SCREENSHOTS



The screenshot shows a 'REGISTER' form with the following fields: Username (filled with 'anshukumar'), Email (placeholder 'Email'), Number (placeholder 'Number'), and Password (masked with '.....'). A 'Register' button is at the bottom, followed by the text 'Already have an account? Login here.' The background features a blue banner with 'Government for Individuals' and a stylized person icon.

Figure B.1 - Register Page



The screenshot shows a 'LOGIN' form with the following fields: Username (filled with 'anshukumar') and Password (masked with '.....'). A 'Login' button is at the bottom, followed by the text 'Don't have an account? Register here.' The background features a blue banner with 'DIGITAL INDIA' and 'Growth is good'.

Figure B.2 - Login Page

<div>Search for Schemes</div>										
SNO	SCHEME NAME	CITIZENSHIP	STATE	AGE	GENDER	CASTE	RATIONCARD	EMPLOYMENTSTATUS	AADHAARNUMBER	LINK TO APPLY
1	Pradhan Mantri Kisan Samman Nidhi	Indian	All	18+	ANY	All	APL / BPL	FARMERS	999999999	Apply For Scheme
2	Atal Pension Yojana	Indian	All	60+	ANY	ANY	Mandatory	Any Field	Mandatory	Apply For Scheme

Figure B.3 - List of Subsidy

Smart building

APPLICATION FORM

First Name

Last Name

Date Of Birth

citizenship

State

Age

Gender

Caste

RationCard No

Employment Status

Aadhaar Number

Scheme Name

Submit

Figure B.4 - Application Form

APPLICATION DETAILS


NAME : BOSS BOSS

DATE OF BIRTH : JAN. 1, 2000

CITIZENSHIP : INDIAN

AADHAAR NO : 9999999999999

SCHEME NAME : PRADHAN MANTRI KISAN SAMMAN NIDHI (PM-KISAN)



SEAL / SIGN

Figure B.5 - Application Details



Figure B.6 - Home Page

APPENDIX-C

ENCLOSURES

1. Research Paper Presented.

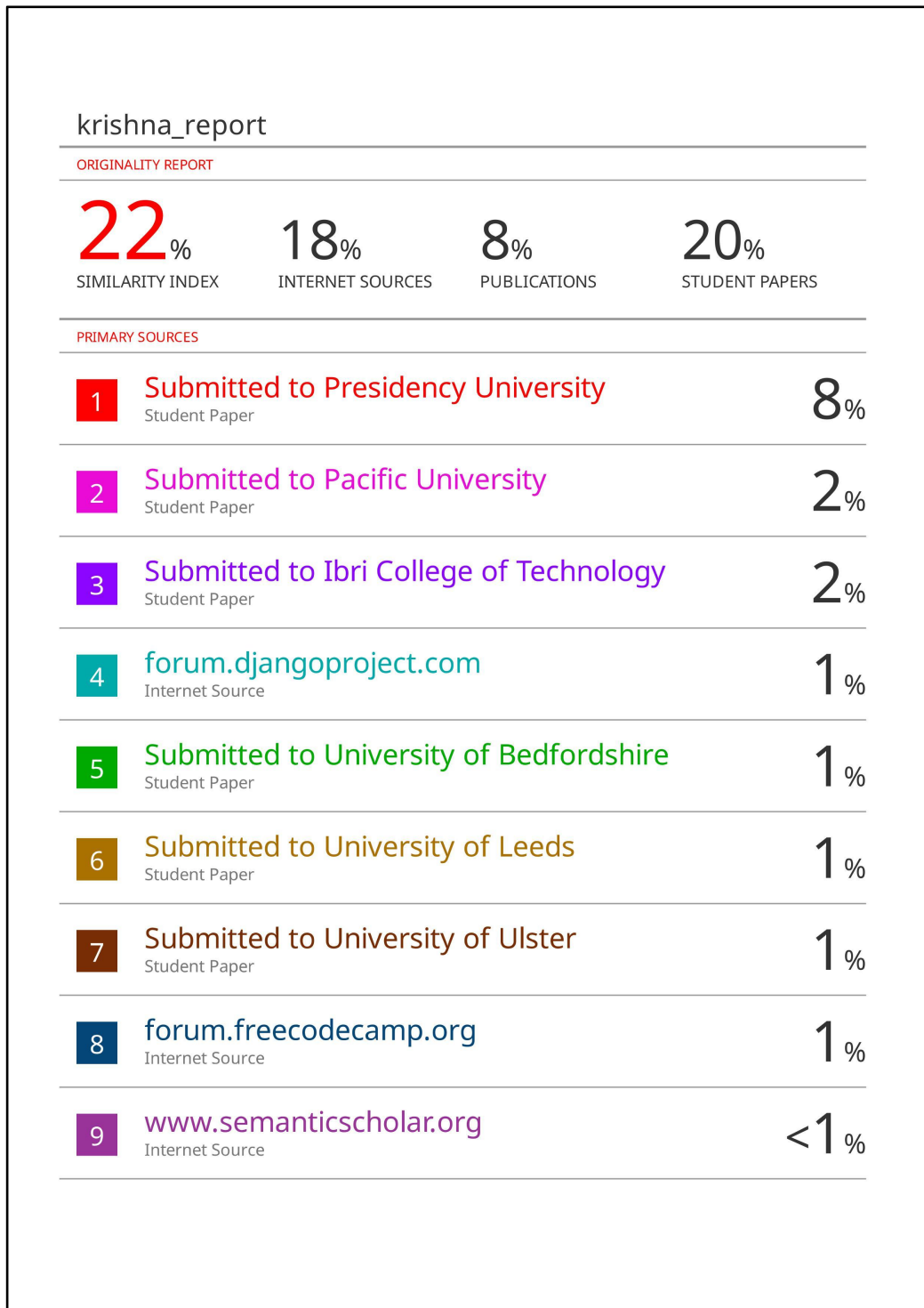
Anshu Kumar, Sreeja Gvs, "Digital India Smart Subsidy System", International Journal of Creative Research Thoughts (IJCRT), Volume 12, Issue 1, January 2024, ISSN: 2320-2882
Available at : <https://www.ijcrt.org/papers/IJCRT2401342.pdf>

2. Include certificate(s) of any Achievement/Award won in any project related event.





3. Plagiarism Report



SDG GOALS



The project work carried out here is mapped to SDG 9: Industry, Innovation, and Infrastructure.

SDG 9 aims to build resilient infrastructure, promote inclusive and sustainable industrialization, and foster innovation. The Digital India Smart Subsidy System Web Portal contributes to this goal by modernizing and optimizing subsidy distribution processes through the use of cutting-edge web technologies. It emphasizes efficiency, accessibility, and transparency in subsidy-related processes, aligning with the broader objective of fostering innovation in the digital space.