MVJ COLLEGE OF ENGINEERING, BANGALORE – 560067

(Autonomous Institute affiliated with VTU, Belagavi)

Department of Computer Science Engineering (Data Science)

A MINI-PROJECT REPORT TITLED

"SurveyGram: Survey at Ease"

Submitted in partial fulfilment for the award of Degree of Bachelor of Engineering in

Computer Science Engineering (Data Science)

Under the Guidance of

PROFESSOR. MARTHA SIMON, ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING (DATA SCIENCE)

Submitted by –

Suren Joshi	**1MJ21CD051**
Vaibhav Kumar P	**1MJ21CD055**
Sumeetkumar katke	**1MJ21CD061**
Pragati Talekar	**1MJ22CD405**

MVJ College of Engineering, Bengaluru
(An Autonomous Institute)
Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi, Recognized by UGC with 2(f) & 12 (B),
Accredited by NBA & NAAC

MVJ COLLEGE OF ENGINEERING, BANGALORE – 560067

(Autonomous Institute affiliated with VTU, Belagavi)

Department of Computer Science and Engineering (Data Science)

<u>CERTIFICATE</u>

Certified that the mini project work titled *"SurveyGram – Survey at Ease"* is carried out by **Sumeetkumar katke (1MJ21CD061), Pragati Talekar (1MJ22CD405), Suren Joshi (1MJ21CD051),** and **Vaibhav Kumar P (1MJ21CD055)** who are confide students of **MVJ College of Engineering, Bengaluru**, in partial fulfilment for the award of Degree of Bachelor of Engineering in **Computer Science and Engineering (Data Science)** of the **Visvesvaraya Technological University, Belagavi** during the year **2023- 2024**. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the mini project report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed by the institution for the said Degree.

Signature of the Guide	**Signature of Head of Department**	**Signature of Principal**
Prof. Martha Simon	**Dr. Shima Ramesh Maniyath**	**Dr. Suresh Babu V.**
Assistant Professor	Head of Department of CSE-CD	Principal
Dept of CSE-CD	Dept of CSE-CD	MVJ College of Engineering Bangalore
MVJ College of Engineering.	MVJ College of Engineering Bangalore -	- 560067
Bangalore – 560067	560067	

<u>External Viva</u>

Name of Examiner   **Signature with Date**

1.

2.

MVJ COLLEGE OF ENGINEERING, BANGALORE – 560067

(Autonomous Institute affiliated with VTU, Belagavi)

Department of Computer Science and Engineering (Data Science)

<u>DECLARATION</u>

We, **Sumeetkumar katke, Pragati Talekar, Suren Joshi,** and **Vaibhav Kumar P** students of sixth semester B.E., Department of Information Science and Engineering, MVJ College of Engineering, Bengaluru, hereby declare that the mini project titled ***"SurveyGram – Survey At Ease"*** has been carried out by us and submitted in partial fulfilment for the award of Degree of Bachelor of Engineering in **Computer Science and Engineering (Data Science)** during the year 2023-2024.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any Degree or Diploma to any other University.

Place: Bengaluru

Date: 30th of August 2024.

	Name	**Signature**
1.	**Sumeetkumar Katke** (1MJ21CD061)	
2.	**Pragati Talekar** (1MJ22CD405)	
3.	**Suren Joshi** (1MJ21CD051)	
4.	**Vaibhav Kumar P** (1MJ21CD055)	

ACKNOWLEDGEMENT

ABSTRACT

Survey Gram addresses the challenge of efficiently connecting companies with potential survey respondents in a mutually beneficial manner. Companies often struggle to gather reliable consumer insights, while users are not always motivated to participate in surveys. Survey Gram offers a solution by creating a platform where companies can post surveys and users can complete them in exchange for rewards. This platform simplifies the process of data collection for businesses and provides users with a straightforward way to earn rewards, thereby fostering higher engagement and quality responses. The project demonstrates the viability of creating an interactive and rewarding survey environment that benefits both companies and users.

Table of Contents

Table of Figure

CHAPTER – 01

INTRODUCTION

Chapter – 01

Introduction

1. Introduction

Survey Gram is a survey platform developed to bridge the gap between companies seeking actionable consumer insights and individuals looking for opportunities to earn rewards by participating in surveys. The platform is designed to simplify the process of survey distribution and participation, making it easier for companies to reach their target audience and for users to discover and complete surveys that interest them. By offering incentives such as cash, coupons, and credits, Survey Gram encourages greater user engagement and ensures that businesses receive high-quality responses. This project was undertaken as a mini project for university coursework, aiming to create a practical solution that enhances the survey-taking experience for both companies and users.

2. Problem Statement

Companies often face significant challenges in collecting reliable and timely consumer insights, which are crucial for making informed business decisions. Traditional methods of gathering survey data can be inefficient, leading to low response rates and poor data quality. On the other hand, users are generally not motivated to participate in surveys due to a lack of tangible rewards or an engaging experience. This disconnect between companies' needs and users' motivations creates a gap that hinders effective data collection and analysis. There is a need for a platform that not only facilitates efficient survey distribution and participation but also incentivizes users to provide thoughtful and meaningful responses.

3. Objective

The objective of Survey Gram is to create a platform that efficiently bridges the gap between companies and survey participants, ensuring that both parties benefit from the interaction. By offering incentives, the platform aims to motivate users to engage with surveys, thereby improving the quality of data collected.

- **Connect** companies with targeted survey respondents.
- **Motivate** user participation through rewards.
- **Enhance** the overall quality and efficiency of survey data collection.

4. Future Scope

Survey Gram has the potential to evolve into a comprehensive platform that not only facilitates survey distribution but also expands its functionalities to cater to a broader audience and more diverse data collection needs. Future developments could focus on enhancing user engagement, improving data analytics capabilities, and integrating advanced technologies to provide a more personalized and efficient survey experience.

- **Expand** the range of incentives to attract a larger user base.
- **Integrate** AI-driven analytics to offer deeper insights from survey data.
- **Personalize** the survey experience by tailoring content to user preferences.
- **Enhance** mobile accessibility to reach a wider audience.
- **Incorporate** gamification elements to further boost user engagement.

CHAPTER – 02

LITERATURE SURVEY

Chapter – 02

Literature survey

Paper – 01: The Impact of Incentives on Response Rates and Data Quality in Online Panels

Authors: E. Singer, M. P. Couper

Published Year: 2008

Brief: This paper explores the effects of cash incentives on response rates and data quality in online survey panels. The authors conducted a study where participants were randomly assigned different cash incentives ranging from $0.50 to $3.00. The findings revealed that offering cash incentives reduced the incidence of trap question failures among qualified respondents, indicating an improvement in data quality. However, the study also found that the incentives had limited impact on overall respondent behaviour, with no clear association with eligibility or break-off rates. This suggests that while incentives can improve certain aspects of survey responses, they do not necessarily enhance overall participation or retention.

Paper – 02: Survey Participation: Motivations and Barriers

Authors: E. A. Blair, S. Burton

Published Year: 2005

Brief: This paper analyses the factors that influence individuals' decisions to participate in surveys. The authors identified key motivators, such as interest in the survey topic and the perceived importance of the survey, as well as barriers like time constraints and privacy concerns. The study provides valuable insights for researchers and survey designers on how to improve participation strategies by addressing these motivators and barriers. However, the paper also highlights the variability in the effectiveness of monetary incentives, which can differ based on the survey context and demographic factors.

Paper – 03: Effects of Incentives on Web Survey Response Rates and Data Quality

Authors: J. M. Bosnjak, M. Tuten

Published Year: 2001

Brief: This study examines the impact of financial incentives on web survey response rates and data quality. The authors found that monetary incentives generally lead to higher response rates, suggesting that financial rewards can effectively motivate survey participation. However, the study also discovered that higher incentives might attract less attentive participants, which could negatively affect data quality. This indicates a trade-off between increasing response rates and maintaining high data quality, as larger incentives may lead to more participation but also to more careless responses.

Paper – 04: The Use of Monetary Incentives in Online Panels

Authors: R. A. Peterson

Published Year: 2004

Brief: This paper reviews the role of monetary incentives in online survey panels, focusing on their effectiveness in increasing response rates. The author concludes that monetary incentives are generally effective in boosting participation, making them a valuable tool for researchers. However, the paper also points out the mixed impact on data quality, as larger incentives can attract participants who are primarily motivated by rewards rather than by providing thoughtful responses. This can lead to challenges in ensuring the quality of the collected data.

Paper – 05: Improving Survey Participation with Gamification

Authors: R. C. Goddard, S. R. Lee

Published Year: 2012

Brief: This paper investigates the use of gamification elements, such as points, badges, and leaderboards, to enhance survey participation. The authors found that incorporating these elements can make surveys more engaging, resulting in higher response rates and improved data quality. Gamification adds an interactive and enjoyable aspect to surveys, encouraging participants to stay engaged. However, the paper also notes that gamification may not appeal to all demographic groups, and there is a potential for the novelty effects to diminish over time, which could reduce its long-term effectiveness.

CHAPTER – 03
THEORY & CONCEPT

Chapter – 03

Theory and Concept

1. Purpose:

The purpose of this project is to develop a platform that addresses the challenges faced by companies in collecting high-quality consumer insights through surveys. Traditional survey methods often suffer from low participation rates and poor data quality, as respondents may lack motivation or face barriers to participation. Survey Gram aims to create a solution that enhances both the quantity and quality of survey responses by offering incentives that encourage user engagement. The platform is designed to streamline the survey process for companies while providing users with a rewarding experience, thereby benefiting both parties involved.

2. Existing System:

The existing systems for conducting surveys, particularly online surveys, often rely on traditional methods that do not effectively engage participants or ensure high-quality data collection. Companies typically distribute surveys through email or online panels, offering minimal or no incentives to respondents. While these methods are cost-effective, they frequently result in low response rates, as participants are not sufficiently motivated to complete the surveys. Furthermore, the data collected may be of questionable quality, as respondents may rush through the survey or provide careless answers due to the lack of engagement.

Disadvantages:

• **Low Participation Rates:** Traditional online surveys often suffer from poor response rates, as participants are not adequately incentivized to complete the surveys.

• **Poor Data Quality:** The lack of motivation can lead to low-quality responses, with participants providing careless or incomplete answers.

• **Limited User Engagement:** Existing systems do not typically offer engaging experiences, which can result in respondents losing interest and abandoning surveys partway through.

• **Inconsistent Effectiveness of Incentives:** While some systems do offer incentives, they are often not tailored to the preferences of the respondents, leading to variability in their effectiveness.

3. Proposed System:

Survey Gram is proposed as an innovative platform designed to overcome the limitations of existing survey systems by focusing on enhancing user engagement and improving data quality. The platform introduces a new approach to survey participation by offering tangible rewards such as cash, coupons, and credits, which are directly linked to user activity. Survey Gram's user-friendly interface and personalized survey experience are designed to attract a wider audience and encourage thoughtful participation. By providing a dynamic feed that displays survey details in an engaging format, the platform ensures that users can easily find and complete surveys that match their interests. Additionally, Survey Gram incorporates mechanisms to monitor and improve data quality, ensuring that the responses collected are both reliable and valuable to companies. Overall, Survey Gram aims to create a mutually beneficial environment where companies can gather high-quality insights, and users are rewarded for their contributions.

CHAPTER – 04
METHODOLOGY

Chapter – 04

Methodology

1. Methodology:

The development of Survey Gram followed a structured methodology, focusing on identifying key modules and defining system requirements to ensure a robust and user-friendly platform. The methodology was divided into several stages, including requirement analysis, system design, module identification, and implementation.

Modules Identified:

1. **User Management Module:**
   - **Purpose:** Handles user registration, authentication, and profile management.
   - **Features:** Secure login/logout, profile updates, and user role management.
   - **Outcome:** Ensures that users can securely access the platform and manage their personal information.

2. **Survey Management Module:**
   - **Purpose:** Allows companies to create, manage, and publish surveys.
   - **Features:** Survey creation, editing, deletion, and scheduling of surveys.
   - **Outcome:** Provides companies with a flexible tool to design and manage their surveys efficiently.

3. **Reward Management Module:**
   - **Purpose:** Manages the allocation and distribution of rewards to users.
   - **Features:** Reward calculation based on survey completion, reward tracking, and redemption options.
   - **Outcome:** Motivates users to participate in surveys by offering tangible incentives.

4. **Survey Participation Module:**
   - **Purpose:** Facilitates user interaction with available surveys.
   - **Features:** Dynamic feed displaying available surveys, survey participation tracking, and response submission.

- **Outcome:** Enhances user experience by providing easy access to relevant surveys and tracking their participation history.

5. **Data Analysis and Reporting Module:**
   - **Purpose:** Analyzes collected survey data and generates reports for companies.
   - **Features:** Data aggregation, filtering, and reporting tools.
   - **Outcome:** Helps companies derive actionable insights from survey data, improving decision-making processes.

6. **Notification Module:**
   - **Purpose:** Sends notifications to users about new surveys, rewards, and updates.
   - **Features:** Email and in-app notifications, customizable notification settings.
   - **Outcome:** Keeps users informed and engaged with the platform.

2. System Requirements:

1. **Hardware Requirements:**
   - **Server**: A robust server with at least 16B RAM, multi-core processors, and SSD storage to handle multiple concurrent users and data processing.
   - **Client Devices**: Users can access Survey Gram via any modern device with internet connectivity, including desktops, laptops, tablets, and smartphones**.**

2. **Software Requirements:**
   - **Backend:** Django framework for backend development, providing a secure and scalable environment.
   - **Frontend:** React framework for a responsive and interactive user interface.
   - **Database:** A well-defined and secure database management system for managing user data, survey details, and responses.
   - **Operating System:** Linux-based server environment for hosting the platform.
   - **APIs:** Integration with payment gateways for reward distribution and email services for notifications. (To be integrated in the future updates)

3. **Functional Requirements:**
   - **User Authentication:** Secure login with encryption to protect user credentials.
   - **Survey Creation:** Intuitive interface for companies to design and publish surveys.

- **Reward Distribution:** Automated system for calculating and distributing rewards based on survey completion.
- **Data Analytics:** Tools for real-time data analysis and report generation.

4. **Non-Functional Requirements:**

- **Scalability:** The system should be able to scale efficiently to accommodate a growing number of users and surveys.
- **Performance:** The platform should provide fast response times, even under heavy load.
- **Security:** Implementation of industry-standard security practices to protect user data and ensure safe transactions.
- **Usability:** The user interface should be intuitive, ensuring a smooth experience for both companies and users.

CHAPTER – 05
SYSTEM DESIGN

Chapter – 05

System Design

1. System Design:

The system design of Survey Gram is structured to ensure scalability, security, and an intuitive user experience. It is divided into two primary aspects: architectural design and detailed design of key components. The system follows a multi-tier architecture, separating concerns to facilitate maintenance and scalability.

2. Architectural Design:

Survey Gram employs a **three-tier architecture** comprising the Presentation Layer, Application Layer, and Data Layer.

- **Presentation Layer (Frontend):**
  - **Description:** This layer handles the user interface and user interactions. Built using the React framework, it provides a responsive and dynamic experience, allowing users to seamlessly navigate the platform on various devices.
  - **Components:**
    - o **User Interface:** Developed with React, it includes components like the survey feed, user dashboard, and profile management.
    - o **State Management:** Utilizes Redux (or a similar library) to manage the application state, ensuring consistent data flow between components.
- **Application Layer (Backend):**
  - **Description:** This layer contains the core business logic and processes user requests, manages data flow, and ensures secure communication between the frontend and the database.
  - **Components:**
    - o **Web Server:** Django serves as the backend framework, handling HTTP requests and coordinating with the database.
    - o **API Layer:** RESTful APIs are implemented for interaction between the frontend and backend, facilitating operations like survey creation, user authentication, and reward management.
    - o **Authentication:** Uses Django's built-in authentication system, along with JWT tokens for session management and secure API access.

- **Data Layer (Database):**
  - **Description:** The data layer is responsible for storing and managing persistent data, including user profiles, survey details, responses, and reward information.
  - **Components:**
    - **Database Management System (DBMS):** Currently Django's default data application SQLITE is used for relational data storage, ensuring data integrity and efficient querying.
    - **Data Models:** The data is organized into tables with relationships defined according to the project's needs, such as Users, Surveys, Responses, and Rewards.
    - **Data Security:** Implementation of encryption for sensitive data and secure access controls to protect against unauthorized access.

3. Detailed Design of Key Components:

- **User Management System:**
  - **Functionality:** Manages user registration, login, and profile updates. Users are assigned roles (e.g., survey respondent, company) that define their access and capabilities within the platform.
  - **Design Considerations:** The system uses JWT tokens for secure session management, ensuring that users can authenticate safely across devices.
- **Survey Management System:**
  - **Functionality:** Enables companies to create, edit, publish, and manage surveys. This component allows for the input of survey questions, setting of rewards, and scheduling of survey availability.
  - **Design Considerations:** Provides a user-friendly interface for survey creation, with validation to ensure completeness and correctness before publication.
- **Survey Participation System:**
  - **Functionality:** Displays available surveys to users based on their interests and history. Tracks user responses and ensures data is submitted correctly.
  - **Design Considerations:** Includes mechanisms for dynamic survey display, real-time progress tracking, and error handling to prevent data loss.
- **Data Analytics and Reporting System:**
  - **Functionality:** Analyzes collected survey data and generates reports for companies. Provides insights into response trends, user engagement, and survey effectiveness.

- **Design Considerations:** Uses aggregation functions and filtering mechanisms to offer customizable reports. Ensures data is processed efficiently, even for large datasets.
- **Notification System:**
  - **Functionality:** Sends alerts and notifications to users about new surveys, reward updates, and important platform news. Supports both email and in-app notifications.
  - **Design Considerations:** Allows users to customize notification preferences. Designed to handle high volumes of notifications efficiently.

4. Security Design:

Security is a critical aspect of the Survey Gram platform. The design includes:

- **Data Encryption:** Sensitive user information, such as passwords and payment details, is encrypted both in transit (using TLS/SSL) and at rest.
- **Access Control:** Role-based access control (RBAC) is implemented to restrict access to sensitive data and functionalities based on user roles.
- **Input Validation:** All user inputs are validated and sanitized to prevent common security vulnerabilities such as SQL injection and XSS attacks.
- **Audit Logs:** All critical actions, such as survey creation, user authentication, and reward distribution, are logged for monitoring and auditing purposes.
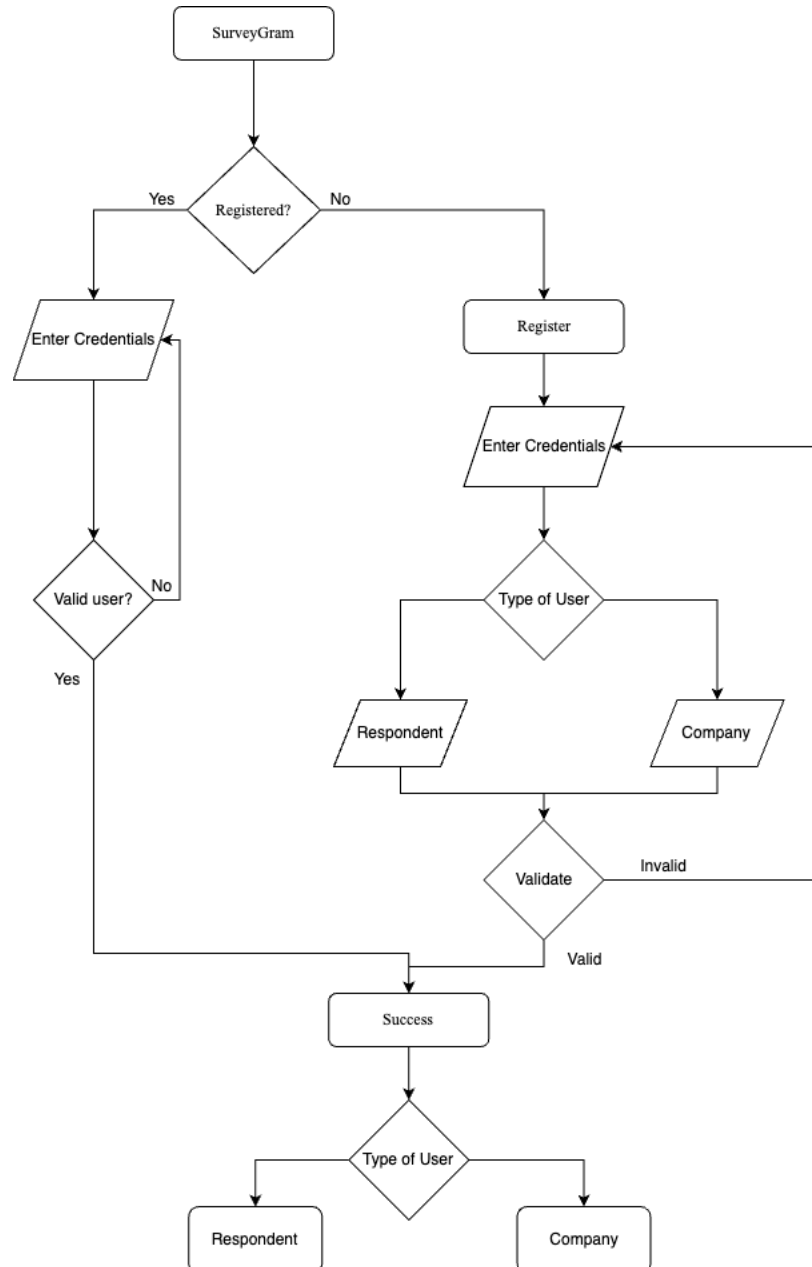
5. Diagrammatic Representation:
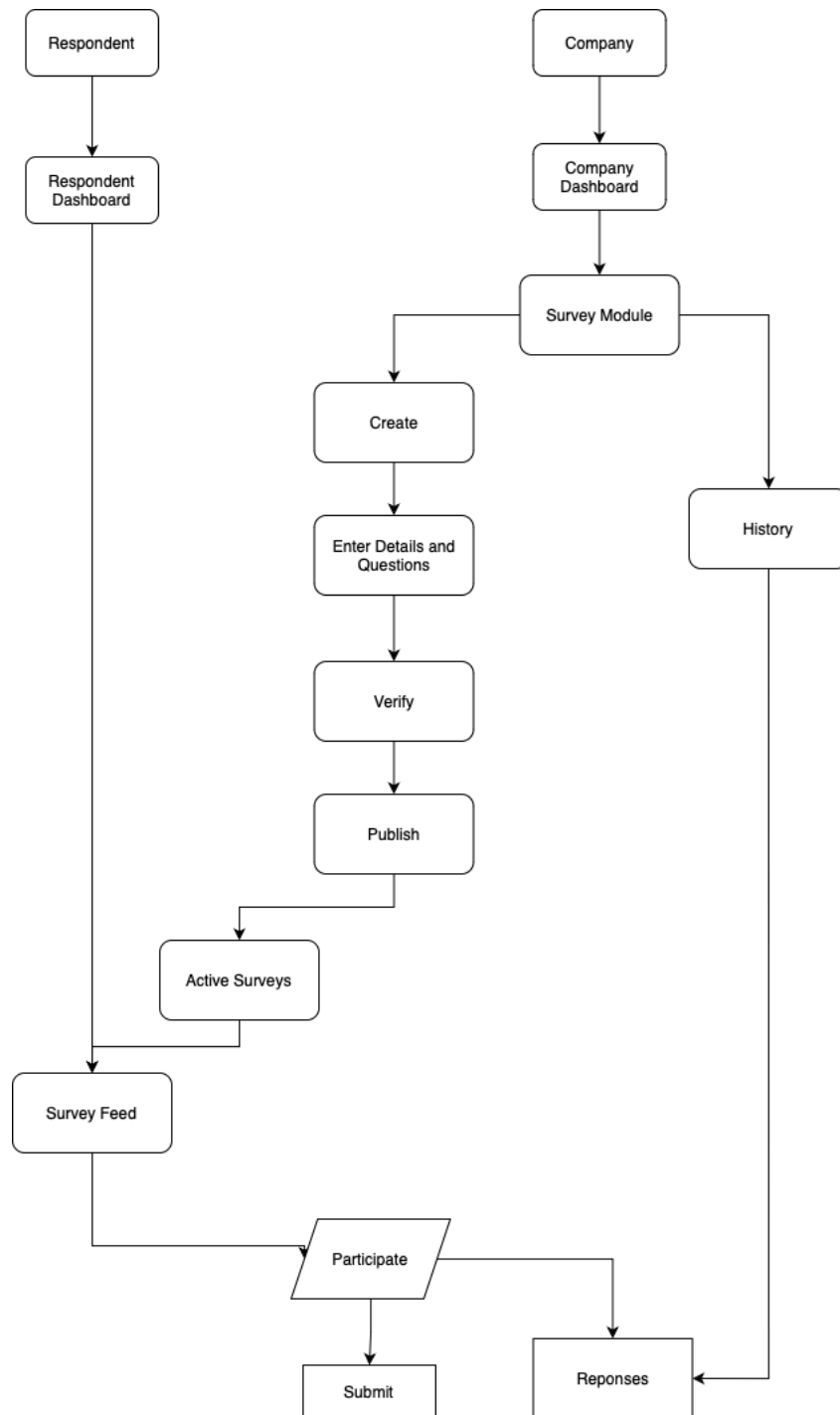
1. Flow Chart:

Figure 1 User Handling Flow

Figure 2 Survey Flow
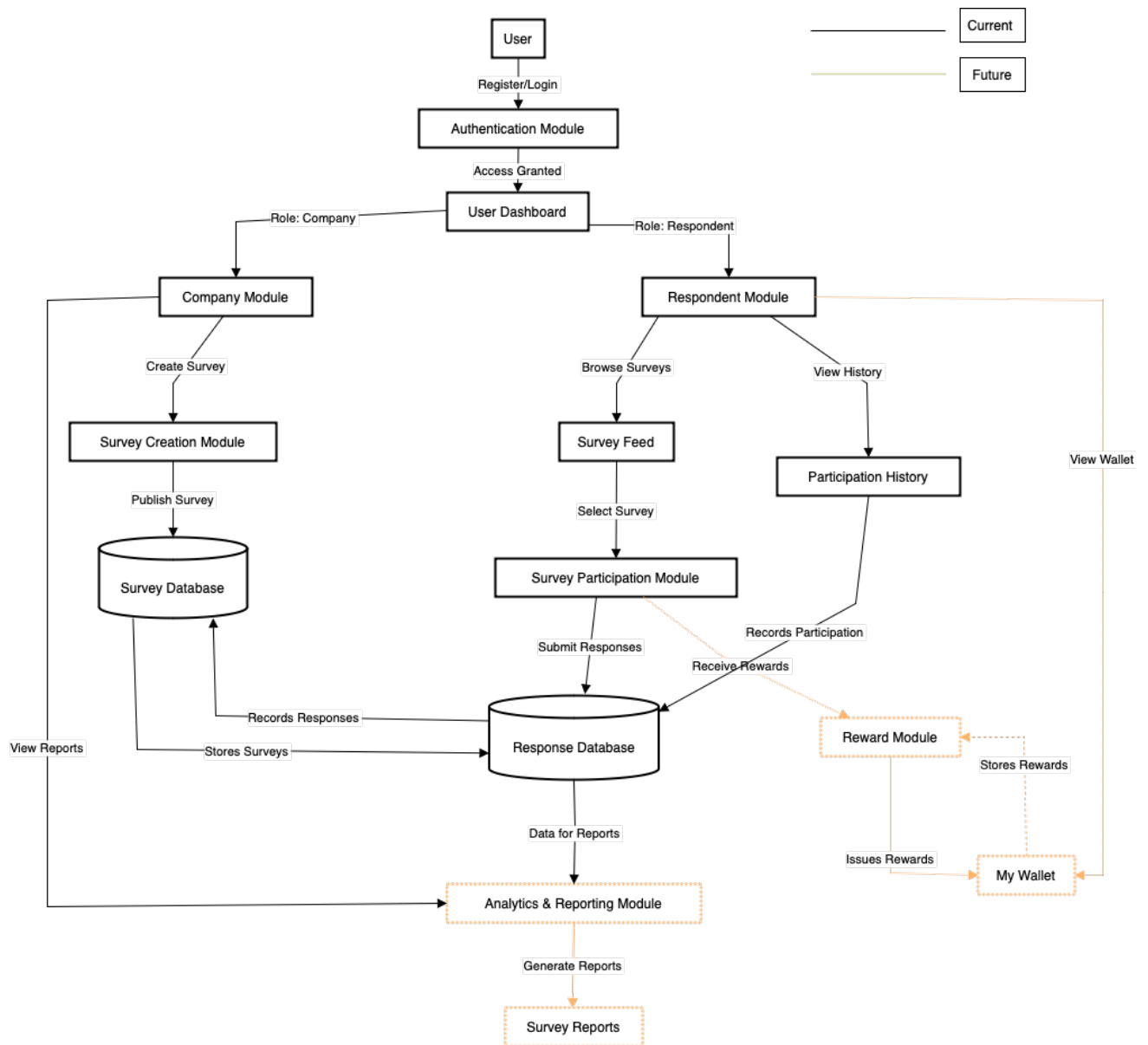
2. System Design:
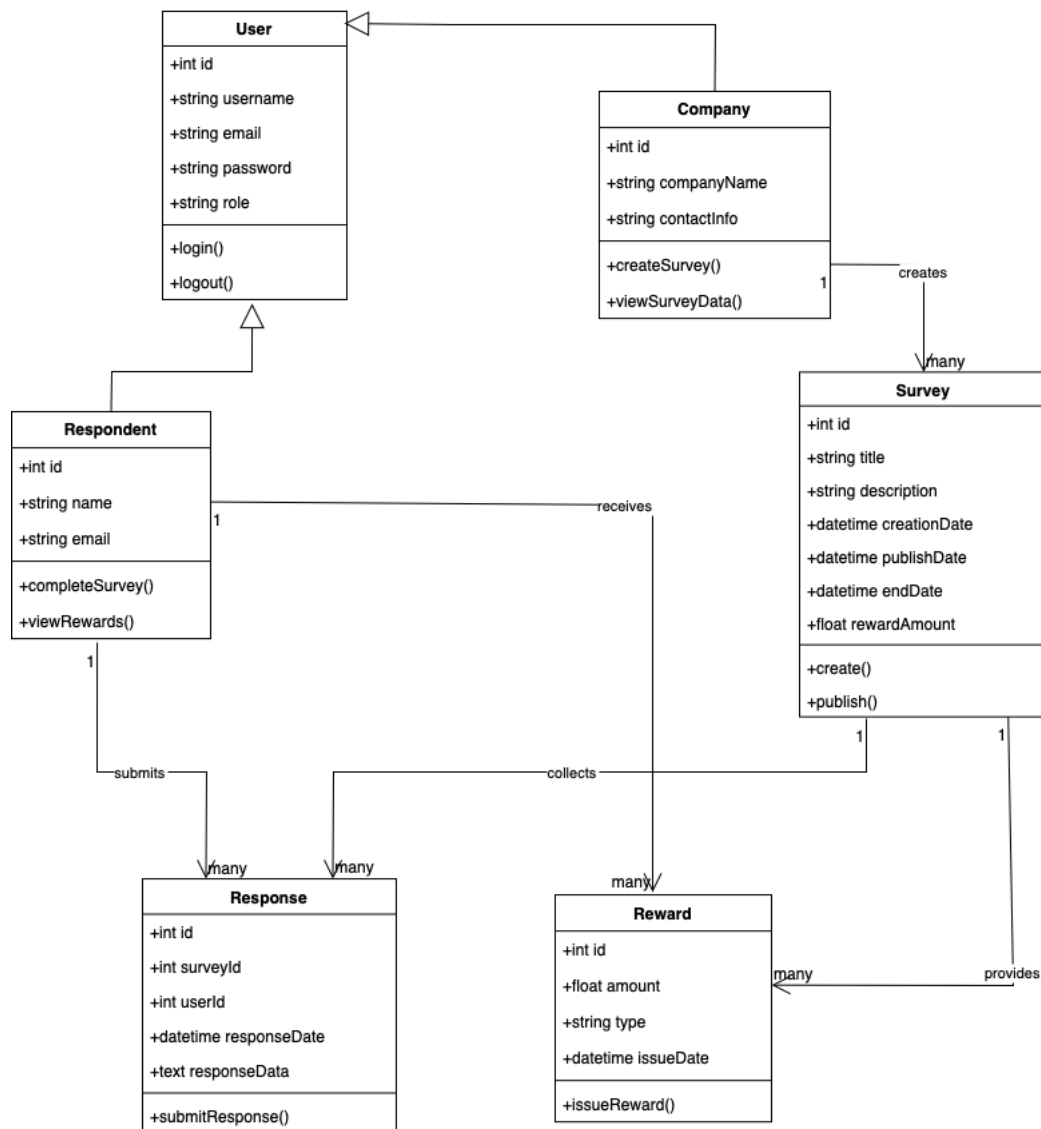

Figure 3 System Design

3. Class Diagram:

Figure 4 Class Diagram
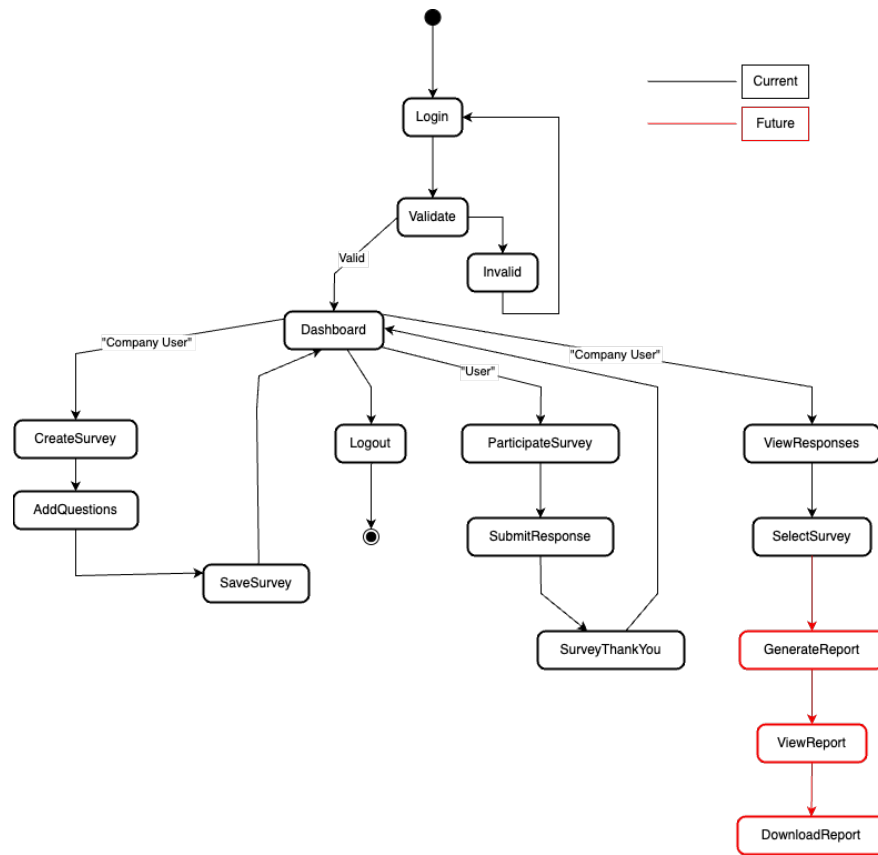
4. Activity Diagram:
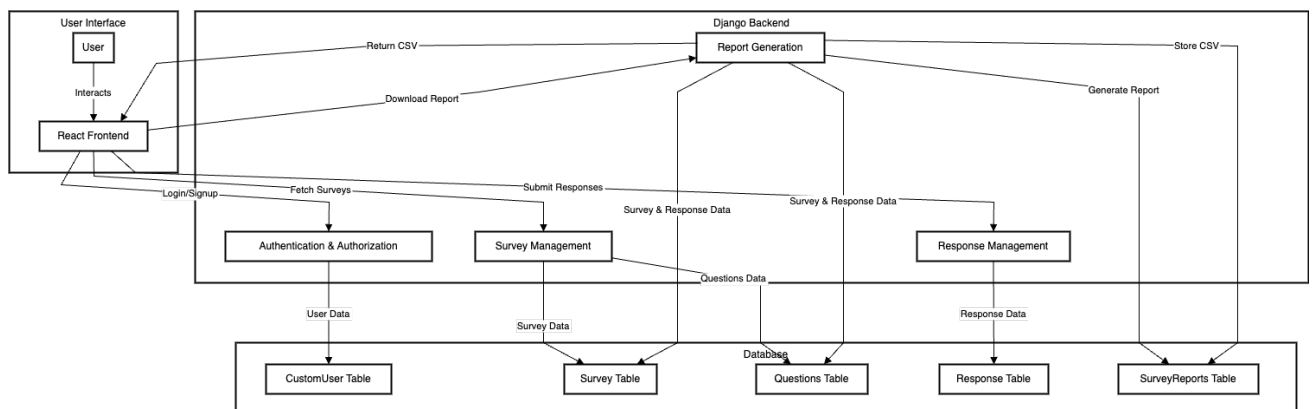

Figure 5 Activity Diagram

5. Data Flow Diagram:

Figure 6 Data Flow Diagram

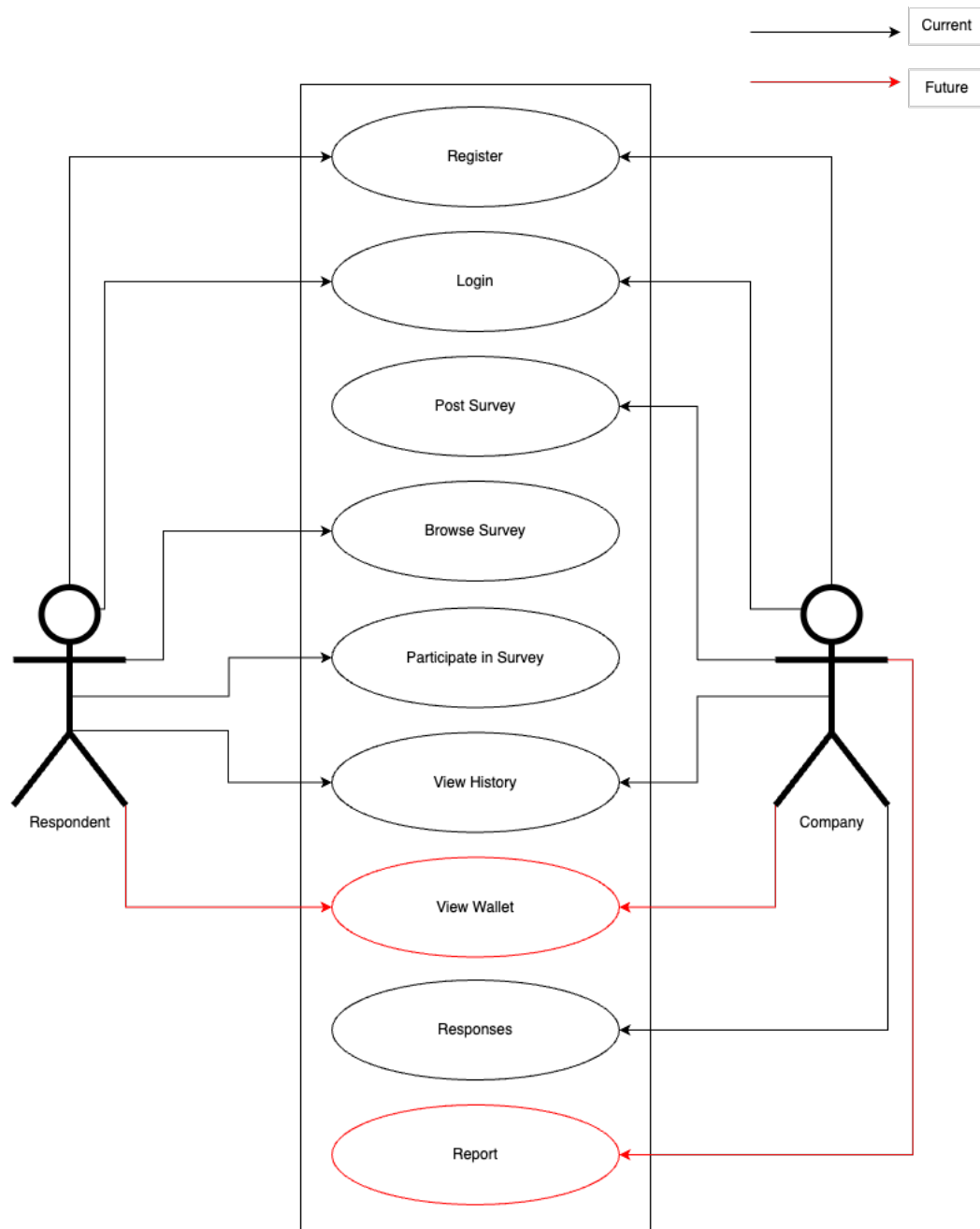## *6.* **Use Case Diagram:**

Figure 7 Use Case Diagram

CHAPTER – 06
IMPLEMENTATION

Chapter – 06

Implementation

1. Overview:

The SurveyGram whose main object is to magnify survey operation which will equally benefit both the company and the respondent aims to deal with higher volume of data handling & manipulation, greater traffic management and provide enhanced user experience which requires a system which possess the capabilities to enable the platform in achieving the named aims.

To ensure that SurveyGram isn't hurdled from achieving its goals, the team came up with the best and most suitable tech stack in building the platform. The whole system as mentioned throughout the report is currently divided into two major segment the company module and the Respondent module.

The platform consists of two parts which complement each other and are very essential for the platform to perform its operation without which it would fail are, the frontend and the backend. The front end is expected to be the face of the project and hence it is required to perform its operations with at most perfection and integrity and hence used Meta's ReactJS, an open-source front-end JavaScript based framework which enable users to build single-page applications. The backend is powered by Django, another open-source, database driven, python-based web-framework which follows a Model-Template-View (MTV) architecture that runs on webserver. We have also employed many other tools, libraries, and third-party components in building the platform.

2. User-Interface:

The user interface is a very crucial aspect of SurveyGram, this becomes a key player in providing the user with one of its kind experiences throughout the process of the survey operations. This section delves into components that make up the user interface, Navigation flow which enable the survey operation, Wireframes and Mock-ups which outlines the base model of the platform user-interface, Responsive design Considerations which removes the device size barrier and enable all type of users to benefit SurveyGram's services.

2.1 User-Component Description:

The team was heavily benefited by the features provided by the selected Front-End framework React Js, which provides reusable component for building the platform. The platform, irrespective of the current module, Respondent or Company, is divided into components such as *Header, Side Navigation Bar* and the *Main Feed* section. We also aim to include a new section *Right plane* which will include some suggestion and tips for the user in the future versions, most likely in *SurveyGram 2.0.*

- **Header and Side Navigation bar:** This component is majorly aims in assisting the user in navigation one through different section of the platform. The Header section helps in operations such as *Login, Logout, Sign-Up,* and interval navigation purposes of the sections such as reward filters and response history etc., whereas the Side Navigation bar, which remains constant in a shrinkable format to achieve responsiveness, aims on providing guidance to the user in moving from one major section such as feeds to another such as rewards, wallet or profile.

- **Survey Cards:** Each survey is presented in a card format, providing users with a snapshot of important details like the company name, registration status, reward type, time to complete, and a brief description. This component is designed to be visually appealing and informative, encouraging user participation. This structure was inspired from several social media platform such as meta's Instagram, Facebook and X corporation's Twitter (now X) ensure in achieving the *social media akin* structure

- **Forms and Input:** The platform includes Form and input components which are the backbone for company user in posting the survey and respondent user in participation. This was carefully designed and build to provide a boredom free experience to the user.

- **Buttons and Call-To-Action (CTA):** Strategically placed button throughout the platform enable in a smooth gliding experience for the users in shifting from one section to another. The CTAs which can be recognised in the form of Post survey buttons, submit response, attain report etc., play a crucial role in performing tasks and invoking backend APIs to execute and provide or provide the data and perform the task

- **Modals and Pop-ups:** For modules requiring additional attention, confirmation and information, modals and pop-ups have been used. These components help in reducing further larger component injection or page loading.

2.2 Navigation Flow:

When the team was building SurveyGram the whole focus was in enabling a confusion less and straightforward approach for the user which will help them in moving through the platform with ease

This was enabled using the *React Routers* which would allow the user to access the desired section without complication the application architecture.

- **Home/Feed:** Upon logging in the user is initially redirected to this section. The respondent can browse the currently active Surveys which he/she can participate in whereas for the company user we aim on providing a *Power Bi* akin data visualization in *SurveyGram 2.0.* We are proud that our fellow teammates Vaibhav Kumar P. and Pragati Talekar are rigorously strategizing the development of this component

- **History:** Entering this page, the user can check their survey history. The company can check their previously posted surveys and if completed can also get a very well detailed report which in the next versions which provide a high-level analysis and for certain type of survey can also get some predictions and more detailed insights. The respondent can browse through their participated surveys and check the status of their reward, we aim on automating this process and provide the rewards as soon as the response is validated and approved.

- **Posting and Participating in Surveys:** Based on the type of user, post-survey or participate in survey form is provided and the user can perform the named operation. SurveyGram intends on ensuring an engaging and boredom-free experience for the users in these operations as they are very crucial and should be handled with outmost attention and care

There exist several other pages too which are designed to provide the user with different details and components. These pages are Rewards/Wallet, Upgrade and Profile which, as the name suggests, provide with respective components and details.

2.3 Wireframes and Mock-ups:

The design process for SurveyGram was guided by the methods of Wireframes and Mock-ups which allowed in getting a clearly picture about the overlook of the user-interface which in turn gave an insight about the complete structure of the platform

This whole assignment was carried out by our skilled team members where Pragati Talekar who possess higher understanding and experience lead the team during the process.

4.1 **Wireframes:** Low-fidelity sketches were initially created to outlined the basic layout and structure of the platform. These wireframes gave a rough understanding about the component layouts and overall navigation flow

4.2 **Mock-ups:** High-fidelity sketches were build using tools such as *Figma* were utilised to implement higher understanding and visualization of the wireframes which implemented the colours, patterns, font, logo design etc., of SurveyGram.

3. Database Design and Management:

This section involves the creation of a logical structure for the SurveyGram database, detailing how the data is stored, managed, retrieved and how the relationships between various entities is defined and managed. The schema design for SurveyGram is tailored to handle complex relationship between user details, company details, surveys and responses. Each entity is represented as a table with clearly defined attributes.

Considering the nature of the platform and sensitivity of the data, the team successfully normalized the database and achieved standards such as *First Normal Form*, *Second Normal Form*, & *Third Normal*. We, by *SurveyGram 2.0*, wish to achieve *Boyce Codd Normal Form, Elementary Key Normal Form* and *Fourth Normal Form.*

3.1 **Tables and Relationship:**

The database for SurveyGram consists of multiple tables, *twelve user declared* and *twenty overall* including the ones provided by the framework used to be specific, each representing different entities involved in the platform's operations. Key tables include:

- **User table:** This can be considered as the base table which contains the general details about the user such as *first name, last name, email, last login, role id, password, is_active, is_blocked etc.* This inherits the Django's user table for

authentication and login which extends the Abstract user table from Django's user models.

- **Role table:** The platform currently contains three types of roles namely, Admin, Company and Respondent. The role_id in the user table is a foreign key from the role table which highlights the type of user. Even though the platform contains an Admin role, we currently don't have an explicitly designed front-end and we tend to use Django's pre-existing admin panel for admin operations.

- **Company Table:** This table is responsible to contain all the company details which are necessary for the verification and tracking of the company user. This extends the user table and has a column *user_id* which is a foreign key from the user table.

- **Survey table:** Contains all the necessary survey details provided by the company such as title, description, and time required etc.

- **Questions table:** Containing a foreign key relation from the survey table this targets to save all the survey questions of the specific survey and options (if exists) in JSON format.

- **Response table:** Possessing foreign keys from user table, survey table and question table, this aims on capturing and storing all the user responses from the survey response form.

There are other tables such as *Type of Question, Reward, Survey History, Survey Reports etc.,* which, as their name suggests, store the designated data which play a vital role in SurveyGram's operation.

3.2 Indexing and Optimization

To ensure optimal performance, indexing is employed on frequently queried columns, such as user IDs, survey IDs, and timestamps. This reduces the time taken to retrieve data and enhances the overall responsiveness of the platform. Additionally, the schema design is optimized to minimize redundancy and improve data access speeds. Query optimization techniques are applied to streamline data retrieval processes, ensuring that the system remains efficient even as the volume of data grows.

3.3 Data Backup and Recovery Strategies

Given the critical nature of the data handled by SurveyGram, robust backup and recovery strategies will be in place. Regular backups of the database will be scheduled to prevent data loss in case of system failures. These backups shall be stored securely using cloud

methods, and a recovery plan will be established to restore data quickly in the event of any disaster. The system will also incorporate mechanisms to ensure data consistency during backups, avoiding corruption or partial saves. This ensures that SurveyGram can maintain its operations smoothly without data integrity issues, even in adverse situations.

4. Backend Implementation:

Backend is a crucial part of SurveyGram, this handles all the data management and processing for the platform. The backend powered by the robustness of Django Software Foundation's python based, open-source Django framework, also compressed with API system which enables a smooth communication between backend and frontend, supported by REST API framework.

4.1 Why Django?

Django is one of the most fastest growing web development server frameworks which provides the user with plenty of libraries and method which takes out the hustle of reinventing the functionality needed.

- **Fast nature:** Django employees a set of reusable components which are built in a very optimised format hence reducing the hustle of recreation which sometimes can be lesser optimised then the predefined ones. It also includes a cashing system which enable in storing frequently used data resulting in lesser number of database queries.

- **Loaded with features:** Django comes in with in-box features such as admin panel, user authentication, site maps, RSS feeds and many more which saves a lot of time and resource while development.

- **Scalability:** Django is very well known for its high scalability nature which matches with our needs perfectly.

- **Security:** Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

4.2 Why REST API?

REST (Representational State Transfer) is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of a distributed, Internet-scale hypermedia system, such as the Web, should behave.

- **Simplicity:** REST APIs are easy to understand and implement, using HTTP methods (GET, POST, PUT, DELETE) for CRUD (Create, Read, Update, Delete) operations.
- **Scalability:** REST APIs can handle large amounts of data and traffic efficiently, making them suitable for high-load applications.
- **Statelessness:** Each request to a REST API is independent, making it easier to scale and maintain.
- **Caching:** REST APIs can be cached to improve performance and reduce server load.
- **Security:** REST APIs can be secured using various mechanisms like authentication, authorization, and encryption.

4.3 API Design and Endpoints:

SurveyGram's API architecture is built considering several use cases and security concerns while focusing majorly on its error handling and logging system which enable in better error handling for the front it.

The APIs are built using several Django libraries such as *status* which helps in packing the response status code*, Response* key function which packs the response for each API call made, validation check methods and other basic python *try-catch*. Several response codes are used for proper indication of type of response which aligns with Web3 status codes.

Each user is provided with their own endpoints for each API call enabling them performing all the request methods (GET, POST, DELETE, & PUT) for the CRUD operations. Some of the endpoints are mentioned below:

- **Common Endpoints:**
  - I.   **Fetching JWT Token [.../token/]:** Extracted from the **J**SON **W**eb **T**oken [JWT] module, returned from *TokenObtainPairView* of JWT this provides

the user with both *access* and *refresh* tokens when provided with valid credentials for authorization

II. **Fetching *refresh* Token [.../token/refresh/]:** Like the prior one, this provides the user with a new *refresh* token whenever the old one expires but only when a valid *access* token is provided.

III. **Login [.../login/]:** As the name says, this handles the login validation of the user and returns the token if authenticated.

- **Respondent Endpoints:**

  i. **Get survey Questions [.../questionFetch/]***:* This fetches the questions from the database for the particular survey; to enable lazy loading this is explicitly called after the user participates in an survey.

  ii. **Fetch Surveys [.../fetch_surveys/]:** This endpoint returns active surveys to the user; the function employs a paging system allowing easy and continuous loading of survey as the user browses. This was inspired from *Instagram's* reel's infinite scroll module, after a thorough study about this module we were able to develop our view.

- **Company Endpoints:**

  i. **Post a survey [.../post_survey/]:** This handles with accepting all the survey data provided by the company user, validated and stores it in the database.

  ii. **Survey History [.../company_history/]:** Handles company user's survey history which return all the past surveys posted by the user.

This whole system is secured using JWT authentication services which provides and robust token-based authentication system and allows only authenticate users to access any of the endpoints mentioned before.

5. Front-End Implementation:

SurveyGram's frontend is built using ReactJS, a powerful and flexible JavaScript library for building user interfaces. ReactJS is chosen for its component-based architecture, allowing for reusable and maintainable UI components. The complete project was built using principles of *stateless component* and *Functions Extending React Components.* Additional libraries, such as Axios for API requests and Redux for state management, are integrated to enhance functionality and improve developer productivity.

A custom *stateContext* library akin functionality was created which assisted in streamlining state management alongside provide several reusable functionalities. This plays a very crucial role in functionality of the components and enables in reducing the delay significantly, while also reducing the number of LOCs, eventually reducing the complexity from front end.

5.1 Integration with Backend:

The frontend and backend of SurveyGram are tightly integrated, with the frontend consuming the backends' RESTful APIs to fetch and display data. This integration is achieved using Axios, which facilitates smooth HTTP requests to the backend endpoints. The frontend is designed to handle asynchronous data fetching, ensuring that the UI remains responsive even when dealing with large datasets or complex operations.

A separate class was created for the Axios class file and wherever needed throughout the React App the instance was called. The file was also responsible to update the refresh token after every *(x-1)* minute, where x = Number of minutes the refresh token expires, which ensured that the app doesn't wait for the token to expire, and user faces unexpected logouts or functionality failure

5.2 State Management:

State management is a critical aspect of frontend development, especially in a dynamic application like SurveyGram. A custom Redux akin **stateContext** state container was built for global state management, ensuring that the application's state is consistent and predictable across different components. This approach simplifies data flow and makes it easier to manage complex state transitions, such as user authentication, survey responses, and UI updates. Few State variables and functions are mentioned below:

- **Variables:**
  - **Authentication:**
    - Access Token
    - Refresh Token
    - isLoggedIn
    - User

- o **UI State:**
    - isLoading
    - isSubmitting
    - activeTab
- **Functions:**
    - o clearError
    - o showErrorMessage
    - o showErrorMessage
    - o tokenLogout

6. Survey Creation and Management:

6.1 Process of Survey Creation

Survey creation in SurveyGram is designed to be an intuitive and streamlined process. Companies can create surveys by defining the survey title, description, target audience, and reward type. The platform guides users through each step, ensuring that all necessary details are captured. The creation process also includes options for setting survey duration, number of questions and a brief description about the survey.

SurveyGram automatically calculates time required for a respondent to finish the survey and based on that the quantity of reward is also automatically decided

Figure 8 Survey Creation

6.2 Adding and Managing Questions

Once a survey is created, users can add and manage questions through a user-friendly interface. SurveyGram supports various question types, including multiple-choice, short answer, Boolean and rating scales. Users can reorder, edit, or delete questions as needed, and the system provides real-time previews to help users visualize the final survey layout.

Figure 9 Adding Questions for the survey

6.3 Survey Deployment and Monitoring

After a survey is finalized, it can be deployed to the target audience. SurveyGram 2.0 aims to offers tools for monitoring survey performance, including response rates, participant demographics, and real-time feedback. Companies can track survey progress through a dedicated dashboard, allowing them to make informed decisions about extending or closing the survey based on the data collected.

The surveys are available for participation only after the start date and ends immediately once the end time is matched. This process is managed by the Django server which also aims to enable report creation once either the desired number of responses are hit, or the survey is closed.

7. Report Profiling: (Future version)

In our upcoming version, *SurveyGram 2.0,* we aim on implementing systems which will provide a thorough insight on the response data. To accomplish this our initial step would be data profiling followed by other means of data analytics for in-depth investigation

7.1. Data profiling.

Data profiling is a method of breaking down the provided data and using visualization methods for better understanding of data quality, type, distribution, Anomalies, Relationships and completeness.

These aspects help the user to examine the responses for better quality, inconsistency, inaccuracy, better data analytics and helps in getting a structed reasonable decision making.

- **Exploratory Data Analysis [EDA]:** Exploratory Data Analysis (EDA) is a crucial step in the data analysis process that involves summarizing and understanding the key characteristics of a dataset. It involves various techniques to uncover patterns, relationships, and anomalies within the data. EDA helps analysts gain insights into the data's distribution, identify outliers, and explore potential correlations between variables. By visualizing and summarizing the data, EDA provides a foundation for further analysis and modelling.
  - o For efficient EDA, we aim to utilized the *ydata_profiling* library for comprehensive reports, coupled with *pandas* and *pandas_visual_analysis* for data manipulation and visualization.

8. User Participation and Response Handling

7.1 Survey Participation Process

Respondents can participate in surveys through a simple and engaging process. After logging in, users are presented with a feed of available surveys. Upon selecting a survey, they can answer questions and submit their responses directly through the platform. The participation process is designed to be quick and user-friendly, encouraging higher completion rates.

Figure 10 Survey Card
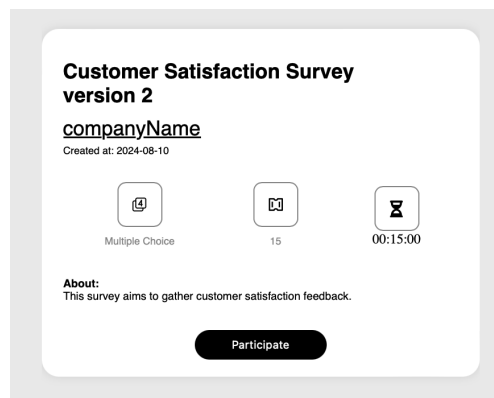
7.2 Submission Handling

SurveyGram ensures that all submitted responses are handled securely and efficiently. Responses are immediately stored in the database upon submission, with confirmation provided to the respondent. The system includes checks to prevent duplicate submissions and ensures that data integrity is maintained throughout the process.
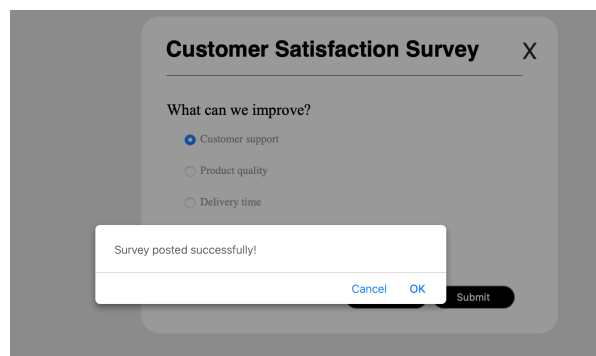

Figure 11 Survey Submission

7.3 Response Validation and Storage

To maintain the quality of data collected, SurveyGram implements response validation mechanisms. These include checks for incomplete responses, inconsistencies, and fraudulent activity. Validated responses are securely stored in the database, with encryption applied to sensitive data to protect user privacy.

9. Security and Privacy Measures:

Security is one of the major principles SurveyGram follows. SurveyGram implements several methodologies to ensure that the user feels safe in using the platform and engaging with the surveys or posting the surveys.

At the same time, SurveyGram ensures that no user access any such data or page which they aren't authenticated for by enabling strict security measures. SurveyGram uses several security features provided by various frameworks and libraries alongside the custom defined security measures.

- **JSON Web Token [JWT]:**

SurveyGram employs JWT based authentication system in the backend for user login and authorization for any operation such as access surveys, post survey, get survey report etc.

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the **HMAC** algorithm) or a public/private key pair using **RSA** or **ECDSA**.
  - **Token:** A JWT Token is divided into three parts
    - o **Header:** which contains the meta data such as algorithm used in encryption,
    - o **Payload:** which contains the claim or data to be transmitted
    - o **Signature:** which is a cryptographic signature that verifies the integrity of the token.

  - **Types of Tokens in Django Auth:** JWT integrated in Django provides it's user with two types of tokens
    - o **Access Token:** This grants temporary access to the resources and has a very short expiry time (5 Minutes in SurveyGram). Once expired the user has to get a new one by authorizing himself

- o **Refresh Token:** This token comes with a very long expiring period (168 hours if checked *Keep me logged in* and 24 hours otherwise) and is used to obtain an access token whenever the access token expires. This can be obtained only by provide valid known credentials.
- **Exemplar Token:**
  - o **AccessToken:**

    eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzI1MDM3Njc2LCJpYXQiOjE3MjQ5Nzc2NzYsImp0aSI6IjRiNTkxNjc3NTViNjQxZjhiMzVkZTYxZmNhNTkyMTFjIiwidXNlcl9pZCI6NCwibmFtZSI6ImNvbXBhbmlOYW1lIiwiZW1haWwiOiJjb21wYW55QHN1cnZleWdyYW0uY29tIn0.jhUPU9jJHVyzEAMZSGIZL9oDeVqJ5oIdeVER1egcF4Y

  - o **Breakdown:**
    - **Header :** { "alg": "HS256", "typ": "JWT"}
    - **Payload:**{"token_type": "access","exp": 1725037676, "iat": 1724977676, "jti": "4b59167755b641f8b35de61fca59211c","user_id": 4, "name": "companyName","email": "company@surveygram.com"}
    - **Signature:** (HMACSHA256(base64UrlEncode(header) + "." +base64UrlEncode(payload), //256-bit-Secret-Key)

      Note: The Token was decoded using jwt.io

10. Summary of Implementation

The development of SurveyGram has been a comprehensive journey that encapsulates the creation of a robust, user-friendly, and scalable platform aimed at revolutionizing the way surveys are conducted and managed. Throughout this report, we have delved into the various aspects of the platform's implementation, covering everything from the initial design and architecture to the detailed development of both the frontend and backend components.

The implementation process began with the establishment of a solid foundation through the design of the database schema, ensuring efficient data management and retrieval. This was followed by the backend development, where a RESTful API was crafted to handle the core business logic, authentication, authorization, and error handling. The frontend was built using ReactJS, focusing on creating an intuitive and responsive user interface that seamlessly integrates with the backend.

Key features such as survey creation and management, user participation, and response handling were meticulously developed to meet the needs of both companies and respondents. The use of state management, particularly through React's StateContext, ensured smooth and efficient handling of various operations across the platform. Additionally, a strong emphasis was placed on security, scalability, and performance optimization throughout the development process.

Overall, SurveyGram's implementation successfully realized the initial vision of creating a platform that not only facilitates easy survey creation and participation but also provides insightful analytics and reports, thereby benefiting both companies and respondents.

11. Reflection on the Development Process

The development process of SurveyGram was both challenging and rewarding, providing invaluable lessons and insights into the complexities of building a full-fledged web platform. One of the most significant aspects of this journey was the collaborative effort that went into the project. Each team member brought their unique expertise, from frontend and backend development to database management and user experience design, contributing to the holistic growth of the platform.

A key takeaway from this process was the importance of adaptability and continuous learning. As the project progressed, new requirements and challenges emerged, requiring the team to adapt and iterate on the initial design and implementation plans. This iterative approach allowed for constant improvements, ensuring that the platform met the evolving needs of its users.

Moreover, the decision to leverage modern technologies like ReactJS and Django proved to be instrumental in achieving the desired functionality and performance. However, it also required the team to stay updated with the latest best practices and updates in these technologies, which was both a learning opportunity and a necessity for the project's success.

The development process also highlighted the significance of thorough testing and quality assurance. By implementing rigorous testing protocols, the team was able to identify and resolve potential issues before they could impact the user experience. This proactive approach ensured that SurveyGram not only met but exceeded the quality standards set at the project's inception.

In conclusion, the development of SurveyGram was a testament to the power of collaboration, innovation, and dedication. The platform stands as a robust solution to modern survey management, poised to make a significant impact on the way surveys are conducted and utilized. As we reflect on this journey, it is clear that the lessons learned and the skills acquired will serve as a strong foundation for future projects, driving continued growth and success in the field of web development.

CHAPTER – 07
CONCLUSION

Chapter – 07

Conclusion

The development and implementation of SurveyGram have been both challenging and rewarding, culminating in a platform that effectively addresses the needs of modern survey operations. From the outset, our objective was to create a system that would enhance the efficiency of survey management while ensuring a seamless user experience for both companies and respondents. This project has not only met these objectives but has also exceeded expectations in various areas.

The implementation phase showcased the integration of advanced web technologies, such as React for the frontend and Django for the backend, resulting in a cohesive and responsive platform. The use of a well-structured state management system, coupled with robust API design, ensured that SurveyGram could handle high volumes of data with ease, while also providing a fluid and intuitive user interface. These technical achievements were complemented by thoughtful design considerations, such as responsive layouts and user-centric navigation flows, which together created a platform that is both powerful and easy to use.

Reflecting on the development process, it is clear that the success of SurveyGram was driven by a combination of meticulous planning, collaborative teamwork, and adaptive problem-solving. The project required a deep understanding of both technical and user-centric aspects, and the team consistently rose to the occasion, overcoming challenges and implementing innovative solutions. This process has not only resulted in a functional and scalable platform but has also provided valuable lessons in project management, software development, and user experience design.

In conclusion, SurveyGram represents a significant achievement in the realm of survey platforms, offering a robust solution that addresses the complexities of survey creation, management, and participation. The completion of this project marks an important milestone, but it also opens the door to future enhancements and expansions. The foundation laid by this implementation provides a solid base for ongoing development, ensuring that SurveyGram will continue to evolve and adapt to the changing needs of its users. This project has been a testament to the power of teamwork, innovation, and a relentless focus on quality, and it stands as a proud accomplishment for all involved.

CHAPTER – 08
REFERENCE

Chapter – 08

Reference

1. Reference Papers:

1. **Singer, E., & Couper, M. P.** "The Impact of Incentives on Response Rates and Data Quality in Online Panels."
   - https://doi.org/10.1177/1525822X12440045

2. **Blair, E. A., & Burton, S.** "Survey Participation: Motivations and Barriers."
   - https://journals.sagepub.com/doi/10.1177/0038038501035004005

3. **Bosnjak, M., & Tuten, T. L.** "Effects of Incentives on Web Survey Response Rates and Data Quality."
   - https://link.springer.com/article/10.1007/s11135-012-9685-8

4. **Peterson, R. A.** "The Use of Monetary Incentives in Online Panels."
   - https://journals.sagepub.com/doi/10.1177/0894439313499322

6 **Goddard, R. C., & Lee, S. R.** "Improving Survey Participation with Gamification."
   - https://doi.org/10.1007/978-3-319-77129-3_3

2. Documentations:

4.1 **Django Official Documentation:** https://docs.djangoproject.com/en/5.1/

4.2 **ReactJS:** https://react.dev/blog/2023/03/16/introducing-react-dev

4.3 **Database Normalization:** https://www.geeksforgeeks.org/types-of-normal-forms-in-dbms/

4.4 **JWT:** https://jwt.io/introduction/

4.5 **RESTful API:** https://docs.github.com/en/rest/using-the-rest-api/getting-started-with-the-rest-api?apiVersion=2022-11-28