A Project Report

On

"Swag Bucks Website"

by

Amit Shah	22SE02ML003
B. Bala Murali Krishna	22SE02ML007
D. Anand Kumar	22SE02ML015

under the mentorship of

Ms. Sneha Saini

Professor, School of Engineering

APRIL 2025

P P SAVANI SCHOOL OF ENGINEERING

P P SAVANI UNIVERSITY

NH NO.: 8, VILLAGE: DHAMDOD, TA. MANGROL, NEAR KOSAMBA, SURAT – 394 125. (GUJARAT).

CERTIFICATE

This is to certify that the Project Report submitted by **AMIT SHAH (22SE02ML003)** to the **P P SAVANI UNIVERSITY** for the partial fulfilment of the subject credit requirements is a bonafied work carried out by the student.

This is to further certify that I have been supervising the Major/Minor Project of **AMIT SHAH (22SE02ML003).**

The contents of this report, in full or in parts, have not been submitted to any other Institute or University for award of any degree, diploma or titles.

Sign of Faculty Mentor :

Name of Faculty Mentor: **Ms. Sneha Saini**

Date:

CERTIFICATE

This is to certify that the Project Report submitted by **BATTULA BALA MURALI KRISHNA KISHORE (22SE02ML007)** to the **P P SAVANI UNIVERSITY** for the partial fulfilment of the subject credit requirements is a bonafied work carried out by the student.

This is to further certify that I have been supervising the Major/Minor Project of **BATTULA BALA MURALI KRISHNA KISHORE (22SE02ML007).**

The contents of this report, in full or in parts, have not been submitted to any other Institute or University for award of any degree, diploma or titles.

Sign of Faculty Mentor :

Name of Faculty Mentor: **Ms. Sneha Saini**

Date:

CERTIFICATE

This is to certify that the Project Report submitted by **DARIVEMULA ANAND KUMAR (22SE02ML015)** to the **P P SAVANI UNIVERSITY** for the partial fulfilment of the subject credit requirements is a bonafied work carried out by the student.

This is to further certify that I have been supervising the Major/Minor Project of **DARIVEMULA ANAND KUMAR (22SE02ML015).**

The contents of this report, in full or in parts, have not been submitted to any other Institute or University for award of any degree, diploma or titles.

Sign of Faculty Mentor :

Name of Faculty Mentor: **Ms. Sneha Saini**

Date:

ACKNOWLEDGEMENT

This report would not have been possible without my teachers who were always there when we needed them the most. We take this chance to acknowledge them and extend my sincere gratitude for helping me make this report a possible.

I wish to thank my faculty mentor **Ms. Sneha Saini,** Professor, School of Engineering. It has been an honor to learn under their mentorship.

As my mentor, he has constantly motivated us to remain focused on achieving my goal. Their observations and guidance helped us to establish the overall direction of the report and to move forward with learning in depth. Their vital support at each juncture, which culminated in successful completion of my project work. I express my sincere gratitude to them for constant support during the project work.

We are also thankful to faculty members of the department for constant support and guidance.

We are thankful to **Dean Prof. Niraj Shah sir**, School of Engineering for his initiative of imparting project during tenure of your study making us learn new things and to help us in expanding our horizons.

Name of Student : **AMIT SHAH (22SE02ML003)**
Name of Student : **B BALA MURALI KRISHNA KISHORE (22SE02ML007)**
Name of Student : **DARIVEMULA ANAND KUMAR (22SE02ML015)**

ABSTRACT

In the evolving digital landscape, Swag Bucks redefines the way users engage with online platforms by transforming everyday interactions into rewarding experiences. Through an innovative system that merges AI-powered smile detection, surveys, and referrals, Swag Bucks offers users a fun and interactive way to earn SB coins, which can be redeemed for exclusive rewards.

At its heart, Swag Bucks introduces a unique earning experience where a simple smile becomes a currency. By detecting and validating smiles in real-time, the platform fosters a sense of joy and engagement. Users can also participate in surveys tailored to their interests and invite friends to join, expanding the community while earning even more rewards.

Beyond just earning, Swag Bucks creates a space where users can track their progress, explore personalized insights, and stay motivated through an intuitive and seamless experience. It is more than just a rewards platform, it is an ecosystem that blends technology, gamification, and human interaction to make online engagement more meaningful, enjoyable, and rewarding.

CONTENTS

6. LIST OF FIGURES

7. LIST OF TABLES

8. INTRODUCTION TO PROJECT

The internet has revolutionized the way people interact, work, and engage with digital content. However, with an overwhelming amount of information and platforms available, keeping users engaged has become a challenge for online services. Swag Bucks is designed to bridge this gap by offering a unique, interactive, and rewarding experience where users earn incentives simply by participating in everyday digital activities.

Swag Bucks introduces an innovative AI-powered smile detection system, allowing users to earn SB coins by smiling at their webcam. This feature transforms a simple facial expression into a rewarding experience, making engagement effortless and enjoyable. Unlike traditional rewards-based platforms that rely solely on surveys or tasks, Swag Bucks adds an element of gamification and AI-driven interaction, making participation both fun and meaningful.

Beyond smile detection, Swag Bucks also integrates a survey-based earning system where users can complete questionnaires tailored to their interests and preferences. This not only helps businesses gather valuable insights but also ensures that users are rewarded for their time and opinions. Additionally, the referral program encourages organic growth by rewarding users who invite their friends to join the platform.

To enhance user experience, Swag Bucks features a dynamic dashboard that provides real-time insights into earnings, activity breakdowns, and engagement trends. Users can track their progress, view recent interactions, and stay motivated through a structured reward system. With a transparent coin tracking and redemption process, the platform ensures fair and secure transactions, giving users confidence in their rewards.

The key focus of Swag Bucks is to make online engagement seamless, interactive, and rewarding. By blending artificial intelligence and user-driven incentives, the platform redefines how people interact with digital content.

9. OBJECTIVES

The main objectives of Swag Bucks are to create a dynamic, rewarding, and user-centric platform that seamlessly blends engagement, gamification, and artificial intelligence. These objectives aim to enhance user experience, encourage interaction, and promote growth within the community.

1. **To Provide an Interactive and Enjoyable Earning Experience**
   Swag Bucks seeks to transform the way users engage with digital platforms by offering a fun, interactive earning system. Users will be rewarded for actions as simple as smiling at their webcam, completing surveys, and referring others, turning everyday activities into enjoyable and rewarding experiences.

2. **To Utilize AI-Driven Engagement**
   A key goal is to leverage artificial intelligence to power the smile detection feature, allowing users to earn rewards through an engaging and intuitive interaction. This technology ensures accuracy and enhances the fun factor by transforming a simple smile into currency. The use of AI also ensures that each user's experience remains unique and personalized, based on their individual interactions.

3. **To Promote Active Participation Through Gamification**
   Swag Bucks aims to encourage continuous user interaction by incorporating gamification elements such as progress tracking, rewards, and milestones. These features motivate users to stay engaged and return to the platform regularly, ensuring they have fun while earning.

4. **To Ensure Seamless and Rewarding User Experience**
   The platform is designed with the user in mind, offering an intuitive interface that makes it easy to track progress, participate in surveys, and redeem earned rewards. The goal is to make sure that all actions, from earning to redeeming rewards, are easy, seamless, and enjoyable.

5. **To Foster Organic Growth Through Referral Systems**
   A central objective of Swag Bucks is to build a growing community by encouraging users to invite friends through a referral program. By rewarding users for expanding the platform's user base, Swag Bucks aims to create a network effect that drives organic growth while ensuring that users feel part of a broader, dynamic community.

6. **To Provide Data-Driven Insights for Continuous Improvement**
   Swag Bucks seeks to enhance its services by gathering and analyzing user behaviour and preferences. This data will help improve the platform, tailor content, and refine user interactions, making the platform smarter over time. The goal is to offer a personalized experience that caters to users' needs and interests, ultimately making the platform more engaging.

7. **To Create a Transparent and Trustworthy Reward System**
   Ensuring transparency and security is crucial. Swag Bucks's coin tracking and redemption system is designed to maintain fairness and prevent fraud. This objective is aimed at building trust with users, ensuring they feel confident in earning and redeeming rewards.

By fulfilling these objectives, Swag Bucks strives to be a leading platform in the world of incentivized digital engagement, making online participation a rewarding, enjoyable, and valuable experience for all users.

10. SYSTEM ANALYSIS

a) Identification of Need

In an age where digital attention is a valuable currency, user engagement has become increasingly difficult to sustain. Users are constantly shifting between platforms, and unless they find genuine value or connection, they often disengage quickly. Traditional reward platforms, while once effective, are now struggling to keep up with changing user expectations and behaviors. Most existing systems rely on predefined actions like completing surveys, watching ads, or referring friends. These approaches are static and transactional, offering the same type of rewards regardless of how the user actually feels or interacts in the moment. There's little to no focus on real-time user behavior, emotional engagement, or fun, spontaneous interactions that reflect the human side of the digital experience.

As technology evolves, users are looking for more meaningful digital interactions, experiences that recognize their presence, their mood, and their individuality. Simply clicking buttons is no longer exciting. There is a clear gap in the market for platforms that reward users not just for what they do, but *how* they do it.
This is where the need for an AI-powered, behavior-based reward system emerges. A platform that can detect a smile, one of the simplest yet most powerful human expressions, and convert it into a reward brings a whole new layer of positivity, novelty, and interactivity to the digital experience.
The need arises from:

- Decreasing attention spans and engagement rates across traditional platforms.
- Lack of emotional or real-time feedback in current reward systems.
- User demand for fairer, more personalized incentives for their time and presence.
- Technological advancements in AI and computer vision that make emotion-based interaction possible and scalable.

This project, inspired by the core idea of Swagbucks but elevated with real-time smile detection, identifies this gap and seeks to fill it by making users feel seen, appreciated, and rewarded not just for completing tasks, but simply for being present, positive, and human online.

b) Preliminary Investigation

A thorough examination of existing reward-based platforms such as JoyWallet, Swagbucks, and PrizeRebel highlighted a consistent pattern: while these platforms effectively offer incentives for completing tasks like surveys, app downloads, or watching videos, they remain largely transactional and static in nature. The user experience is typically built around third-party content, with little to no real-time interaction or personalized engagement.

These platforms often rely on external survey providers and operate within rigid frameworks. Users are expected to complete lengthy forms or offers, which can become repetitive and disengaging over time. There is minimal integration of real-time feedback mechanisms, emotion-driven responses, or behavior-based reward logic, limiting the potential for deeper user connection or spontaneous participation.

The concept behind this project was shaped through this investigation. By identifying what these platforms lacked, the idea emerged to integrate AI technologies like facial expression detection, live analytics dashboards, and automated reward distribution into a cohesive system. This would provide a more immersive and dynamic experience for users.

Early-stage user feedback and informal testing further validated this direction. Participants expressed strong interest in gamified, low-effort earning mechanisms that felt fun and intuitive, such as being rewarded just for smiling. This insight confirmed the opportunity for a web-based reward system that is interactive, emotion-aware, and rewarding in real time, setting it apart from conventional alternatives.

c) Feasibility Study

Let's break this down into 4 types of feasibility:

- **Technical Feasibility**
  The project is technically feasible using web development technologies like React, Node.js, Firebase, and face-api.js. All required tools and APIs are accessible and well-documented.

- **Economic Feasibility**
  As a college-level project, development costs are minimal. A free-tier deployment on platforms like Firebase or Render helps reduce hosting and storage costs, making it economically viable.

- **Operational Feasibility**
  The system is easy to operate and requires minimal training. The user interface is simple and intuitive, ensuring ease of use even for first-time users.

- **Schedule Feasibility**
  The development was divided into phases—planning, design, development, testing, and deployment—each achievable within the project timeline using proper scheduling tools.

d) Project Planning

The entire project was planned using the SDLC model, with development split across several milestones:

1. Requirement gathering and analysis
2. UI/UX design
3. Backend setup and database structure
4. Smile detection integration
5. Reward system implementation
6. Dashboard & admin analytics
7. Testing & bug fixing
8. Final deployment

Each stage was tracked for progress and dependencies to ensure timely delivery.

e) Project Scheduling

i. PERT Chart:

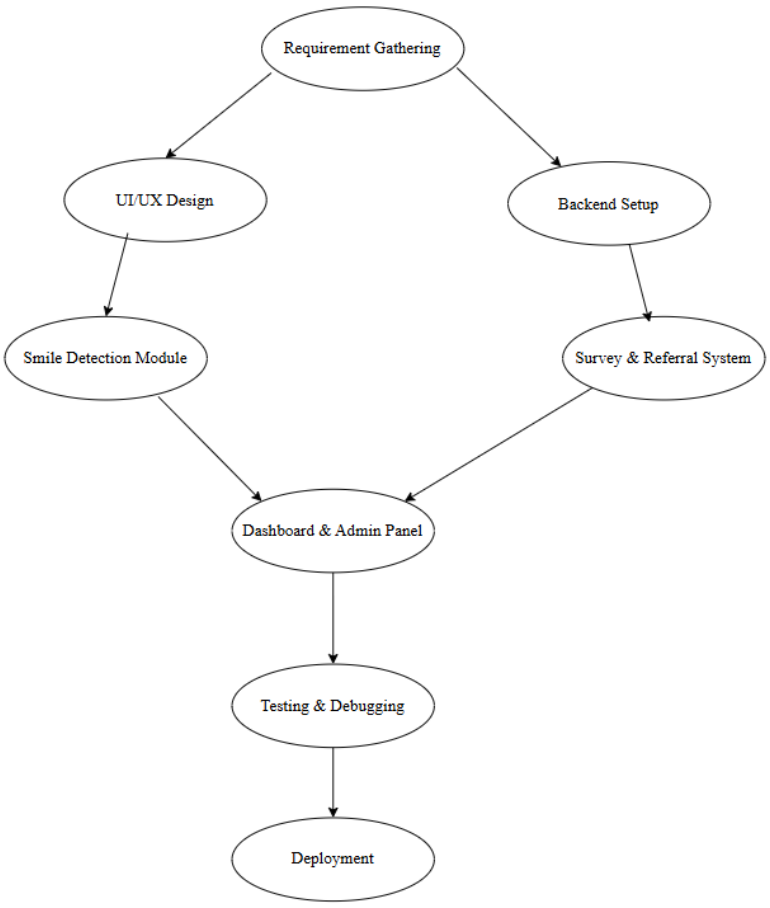


Figure 10.1 PERT Chart

ii. Gantt Chart:

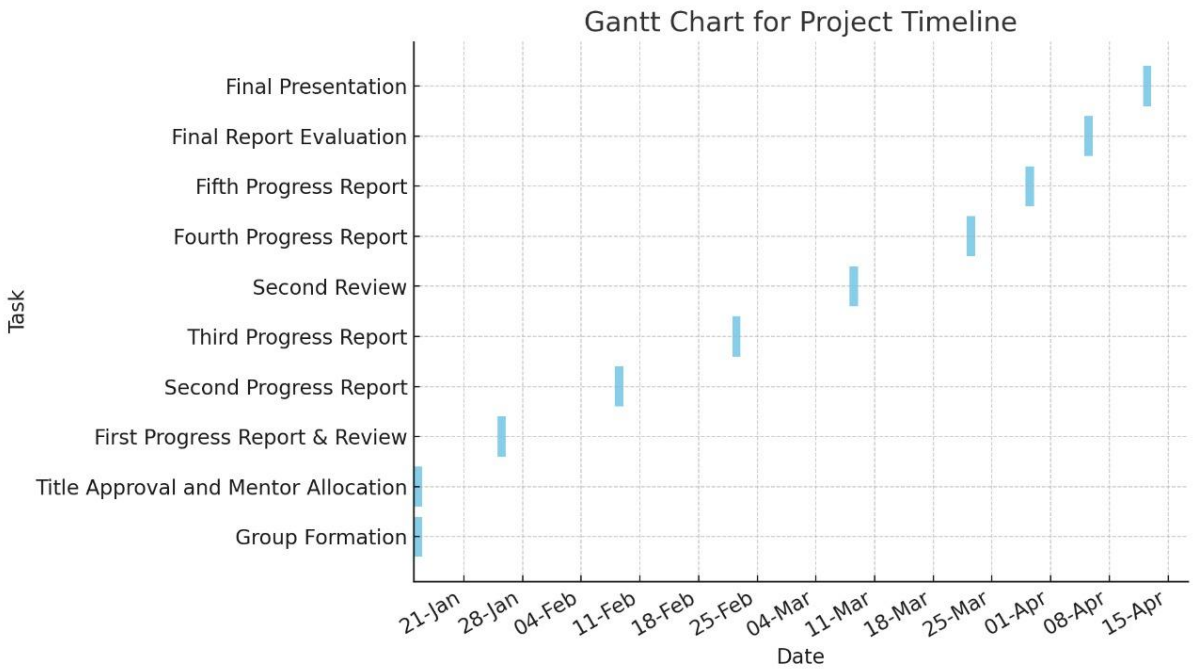


Figure 10.2 Gantt Chart

f) Software Requirements Specifications (SRS)

This Software Requirements Specification (SRS) outlines the design and implementation goals for **Swagbucks**, a web-based reward platform that enables users to earn virtual coins through smile detection, survey completion, and referral-based interactions. The platform focuses on delivering a dynamic and interactive user experience by integrating real-time facial expression recognition and automated reward mechanisms.
Swagbucks allows users to register and log in securely, ensuring proper authentication and data protection. Once authenticated, users can engage in activities such as smiling in front of their webcam, detected using face-api.js—completing surveys, or referring friends to the platform. Each action is tied to a reward system that updates the user's coin balance in real time, providing immediate feedback and motivation.

The platform also features a personalized dashboard where users can track their activity, coin earnings, and behavioral insights. On the administrative side, an admin panel is available to monitor user activity, manage content, and analyze platform-wide engagement patterns. From a functionality standpoint, Swagbucks is built with responsiveness in mind, ensuring a consistent user experience across devices and screen sizes. The underlying system is designed using a serverless architecture to support scalability and cost-efficiency, with a focus on maintaining high performance and low latency, especially in processing webcam-based smile detection. Security is a critical consideration throughout the system, including secure data storage and role-based access control. All personal information and reward data are stored securely, leveraging modern encryption and database practices.

In terms of infrastructure, the platform requires users to have a camera-enabled device (either mobile or desktop) and an active internet connection. Hosting is handled via platforms like Firebase or Render, supporting seamless backend operations. Technologically, the frontend is developed using React.js, while the backend logic is handled through Node.js or Firebase Functions. For persistent data storage, MongoDB is used. The smile detection feature relies on face-api.js, a lightweight and powerful facial recognition library that enables real-time emotion detection through the user's webcam.
This specification serves as a foundational document to guide the development, deployment, and evolution of the Swagbucks platform, ensuring that all requirements, both functional and non-functional, are addressed effectively.

g) Software Engineering Paradigm applied

The project follows the Incremental Model, a type of iterative software development lifecycle, where features were developed and delivered in parts.
Each module (smile detection, dashboard, surveys, referral system) was built, tested, and improved in iterations based on feedback. This allowed flexibility in design, fast debugging, and better control of evolving requirements.
This model was chosen due to:

- Clear module separation.
- Faster delivery of working software.
- Easy testing and integration of components.
- Continuous validation through user feedback.

h) Data Models

1) User Flow Diagram

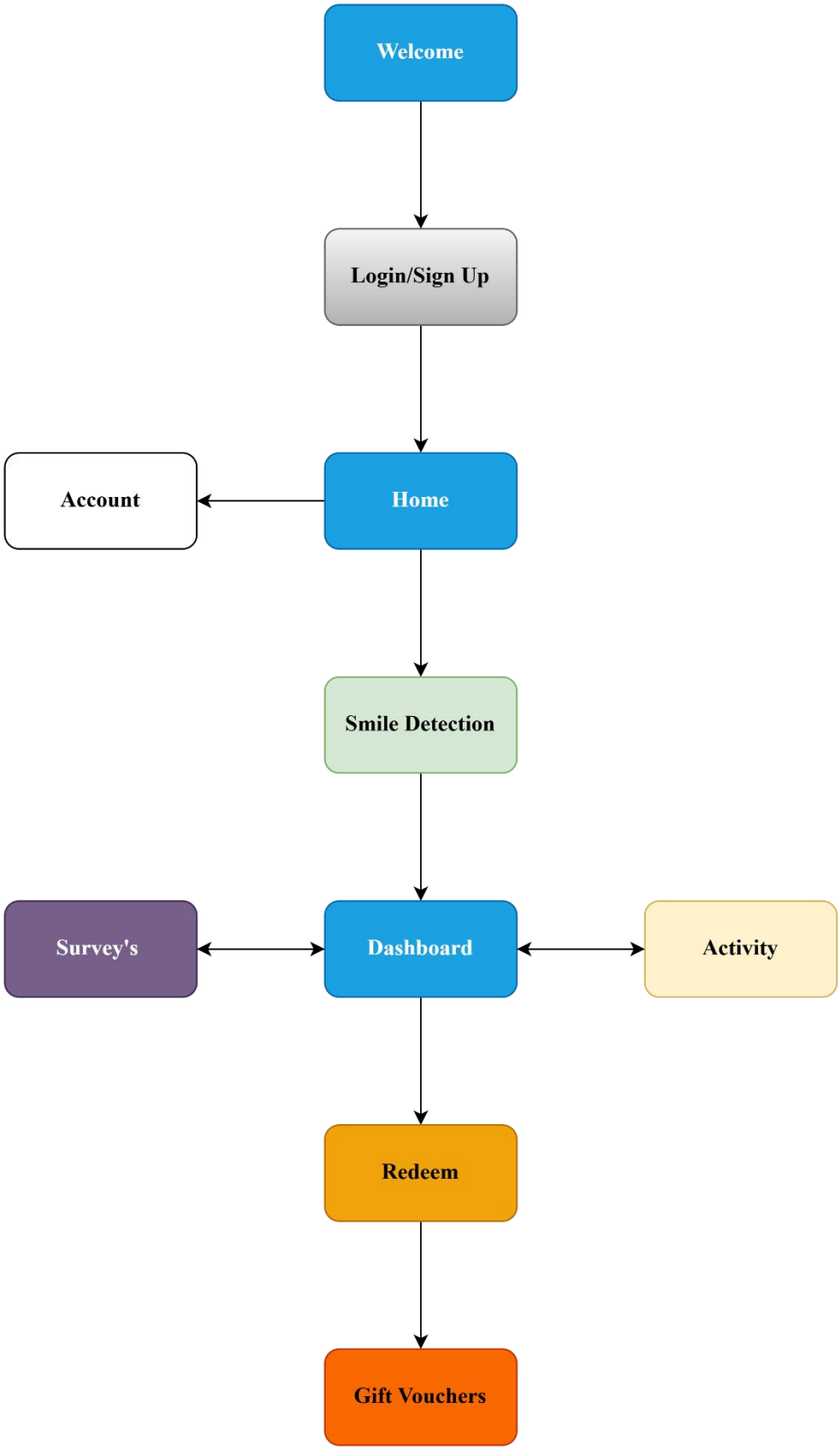


Figure 10.3 UFD

2) Data Flow Diagram (Level 0)

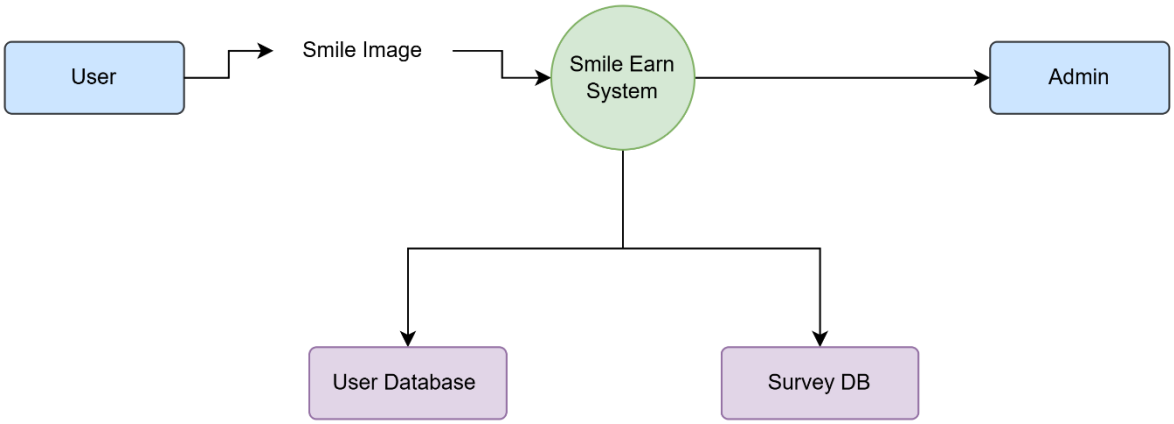


Figure 10.4 DFD (Level 0)

Data Flow Diagram (Level 1)

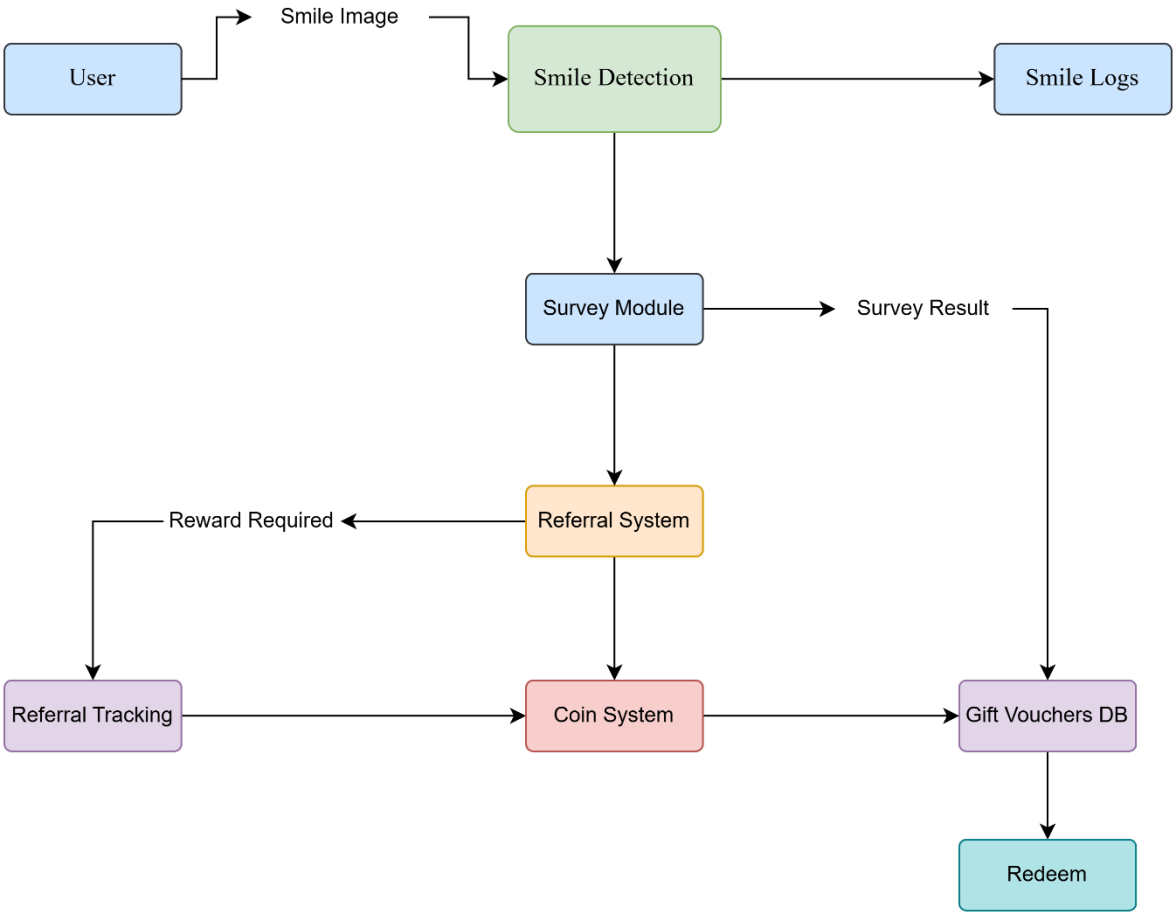


Figure 10.5 DFD (Level 1)

11. SYSTEM DESIGN

a. Modularisation Details

The project is divided into independent functional modules, each responsible for specific tasks to ensure scalability, code reusability, and ease of maintenance.

Modules:

1. **User Authentication Module:** Handles login, registration, and access control.
2. **Smile Detection Module:** Uses the webcam to detect smiles via face-api.js and triggers reward mechanism.
3. **Survey System Module:** Displays surveys and tracks completions to credit SB Coins.
4. **Referral Module:** Generates referral links and tracks invited users.
5. **Rewards & Coins Module:** Tracks SB coin earnings from all activities.
6. **Admin Panel Module:** Allows monitoring, analytics, and control of user activities and content.
7. **Dashboard Module:** Displays live user stats, recent selfies, earnings breakdown, and graph analytics.

Each module is designed to work both independently and collaboratively, following a loosely-coupled, highly-cohesive structure.

b. Data Integrity and Constraints

To maintain accurate and secure data, several constraints and validations are implemented:

- **Unique Constraints:** Email, Username, and Referral Code must be unique.
- **Not Null Constraints:** Fields like UserID, Smile Count, and Survey Status cannot be null.
- **Validation Rules:**
    - Password strength validation on sign-up.
    - Image size and face visibility checks during selfie upload.
    - Duplicate survey attempt checks.

Data integrity is further supported by:

- Transactional operations for coin updates.
- Role-based access (e.g., only admins can edit reward values).
- Referenced keys to maintain relational consistency (e.g., UserID in SmileTable must exist in UsersTable).

c. Database Design, Procedural Design/Object oriented Design

Database Design

The database is designed using a NoSQL structure (MongoDB), but can also be adapted to relational DBs like MySQL.

Key Collections / Tables:

- **Users** (UserID, Name, Email, Password, ReferralCode, Coins)
- **Selfies** (SelfieID, UserID, Timestamp, SmileScore)
- **Surveys** (SurveyID, Question, Options, Reward)
- **Survey Responses** (ResponseID, UserID, SurveyID, Answer)
- **Referrals** (ReferrerID, ReferredID, Timestamp)
- **AdminLogs** (LogID, AdminID, Activity, Timestamp)

Procedural/OOP Design

- Frontend: Modular React components handle different pages like Dashboard, Survey, and Rewards.
- Backend: Node.js APIs are written as reusable service-based functions.
- Smile detection logic is encapsulated into a separate utility that handles webcam inputs and detects expressions in real-time.

d. User Interface Design

The UI is crafted for ease of use, accessibility, and engagement, especially on mobile and desktops.

Design Highlights:

- Clean and modern UI using Tailwind CSS and React.
- Fixed navigation bar with user-friendly icons.
- Dashboard features charts, coin stats, and recent selfies.
- Survey page with clickable, responsive MCQs.
- Reward redemption and referral sections are intuitive.
- Admin panel has filters, analytics, and user stats view.

UX decisions were guided by simplicity and gamification to make earning fun and effortless.

GUI's

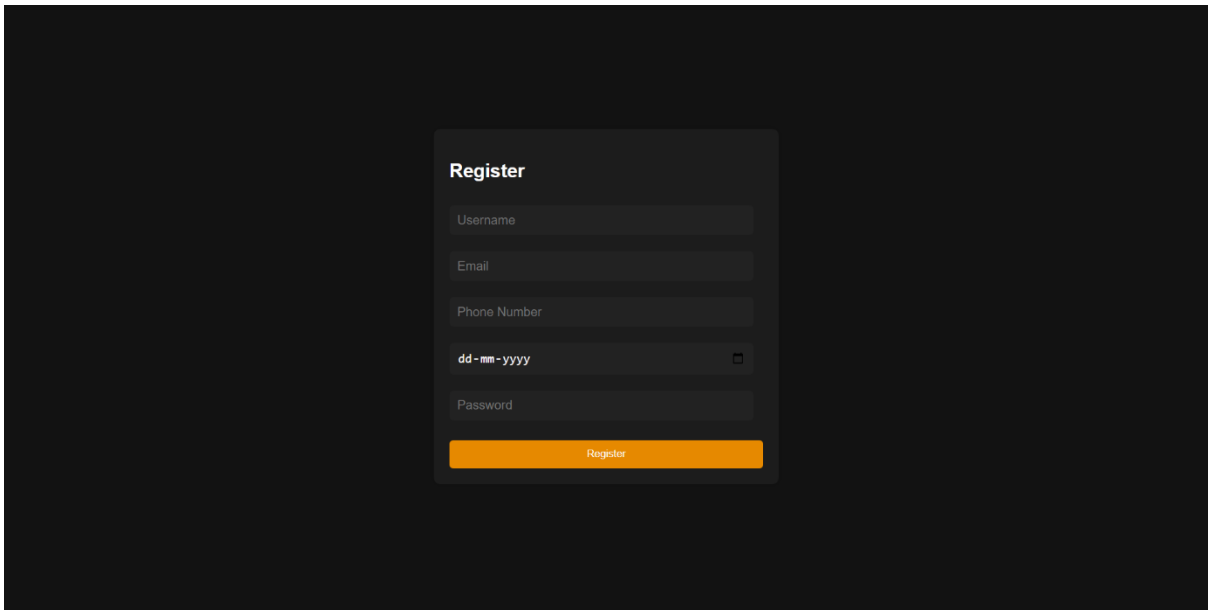


Figure 11.1 User Registration

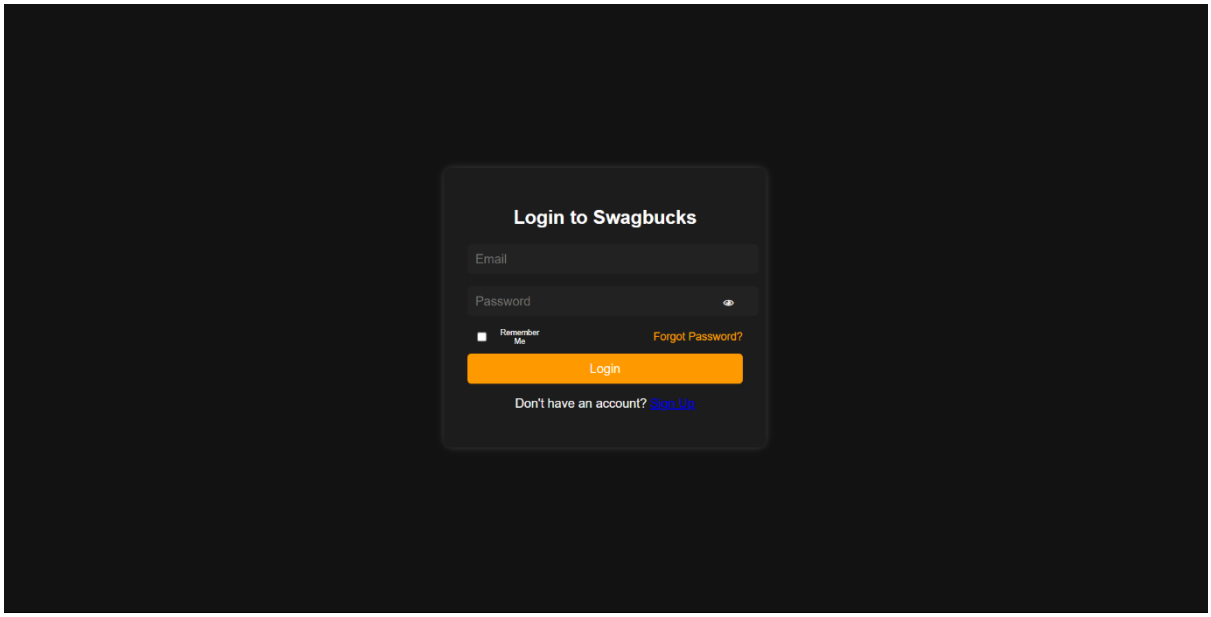


Figure 11.2 User Login

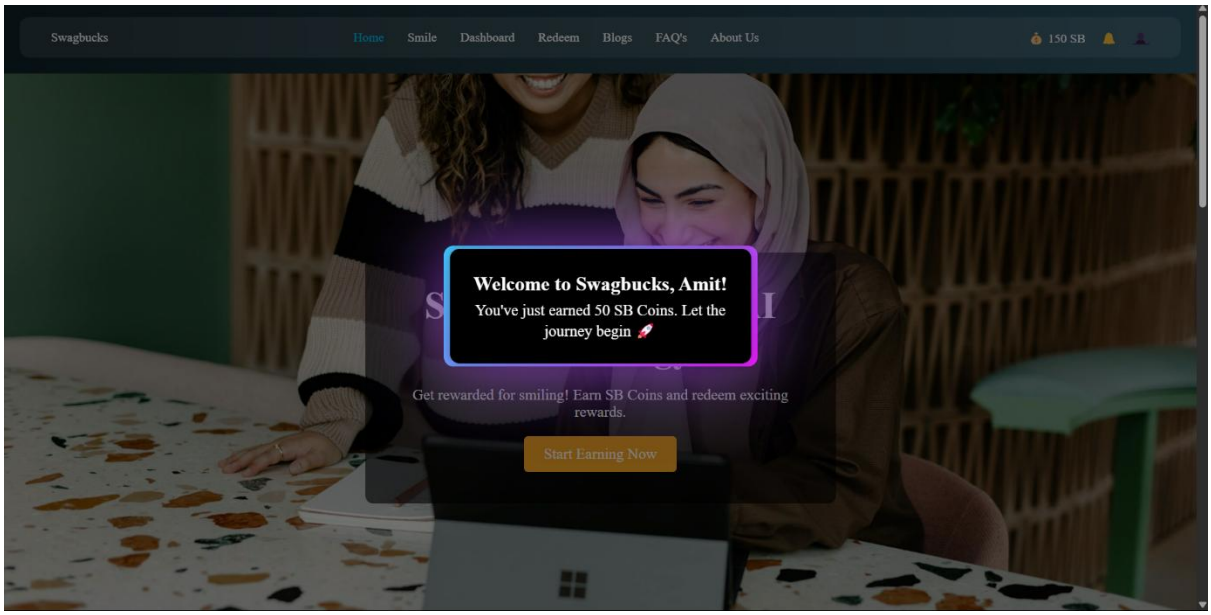


Figure 11.3 Bonus

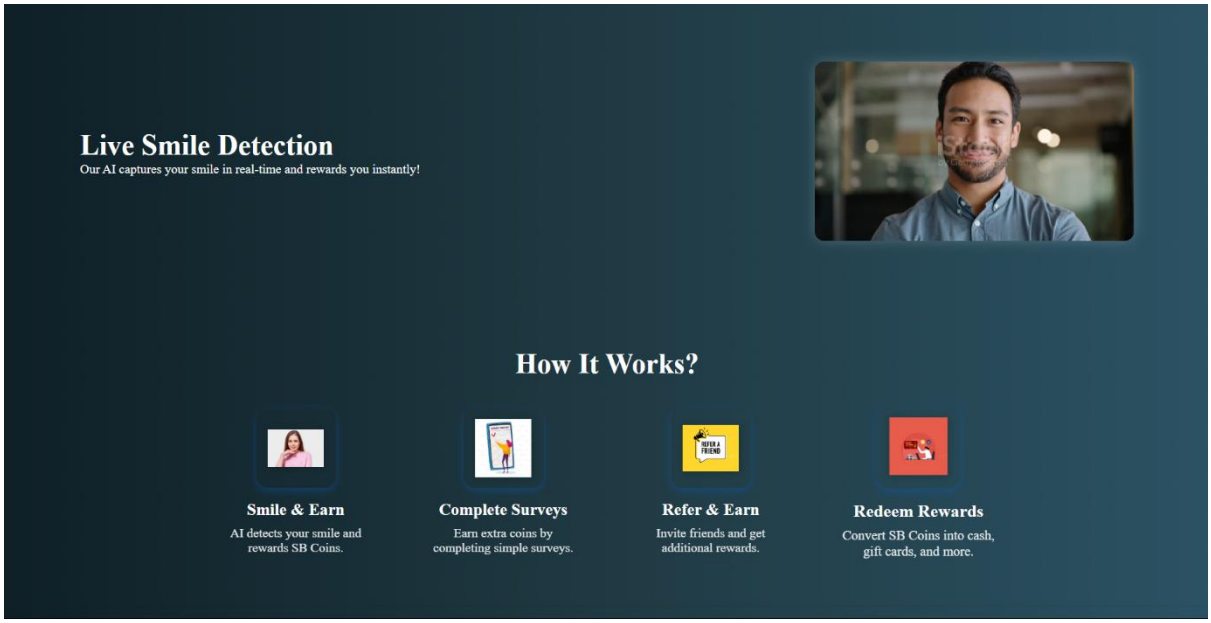


Figure 11.4 Home Page

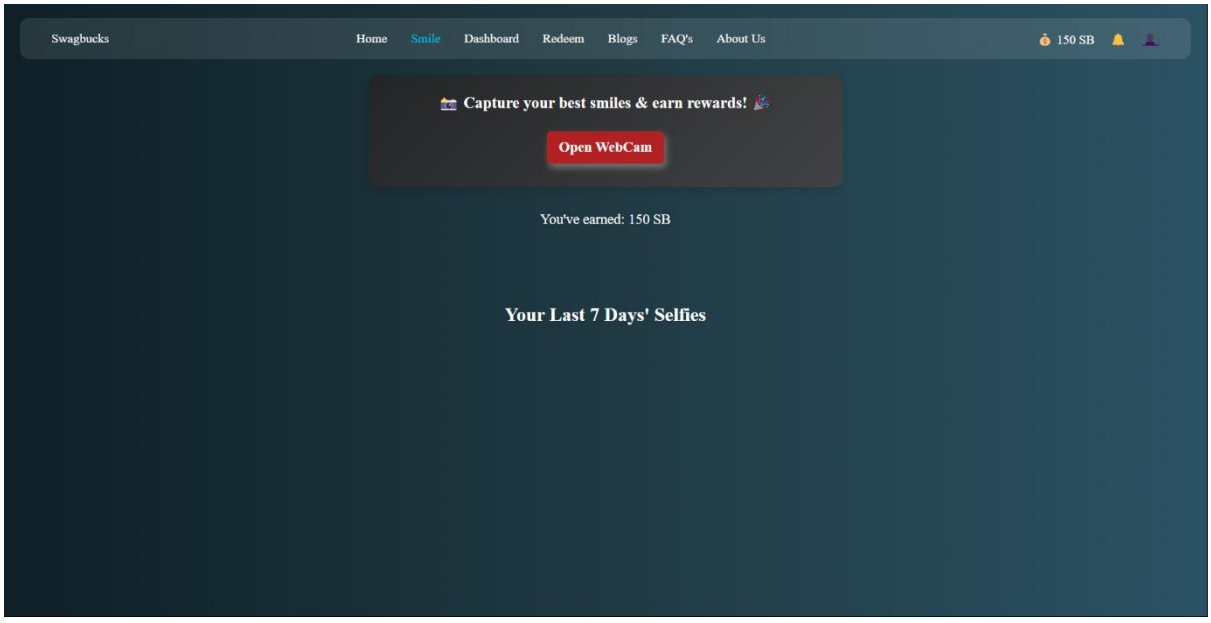


Figure 11.5 Smile Page

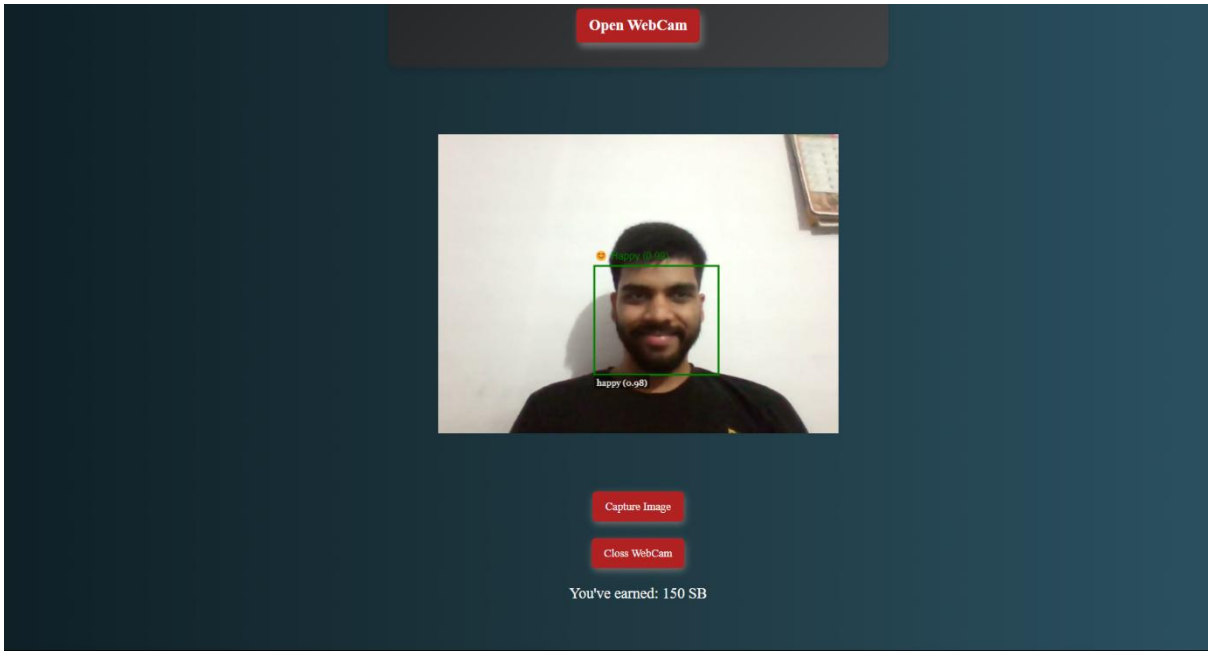


Figure 11.6 Smile Detection

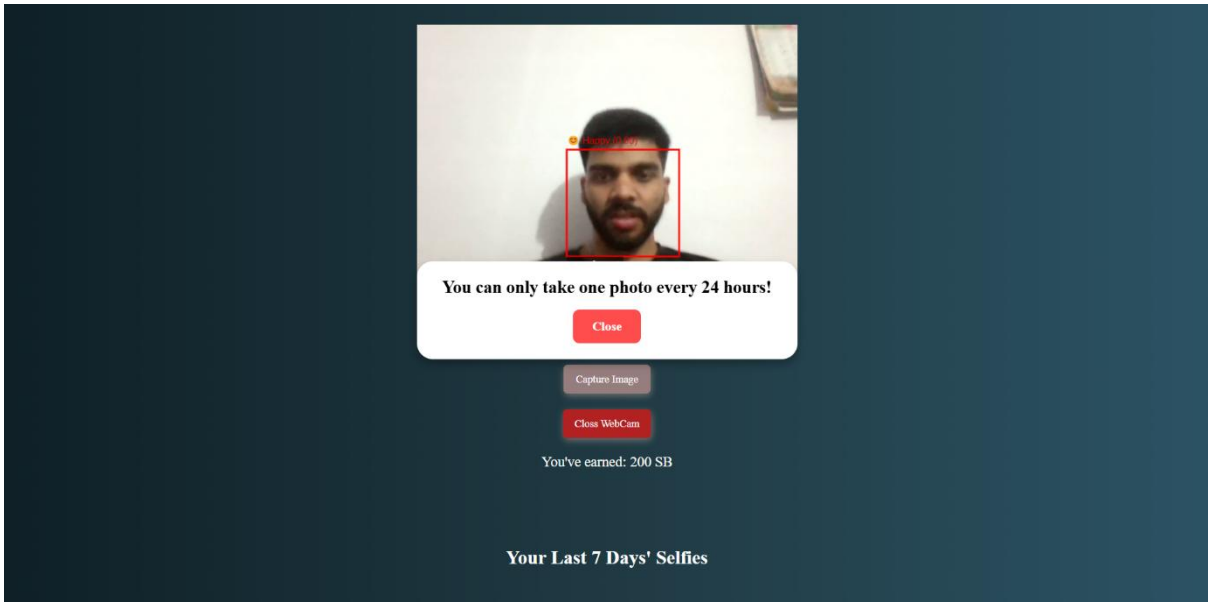


Figure 11.7 Capture Limit

Figure 11.8 Activity

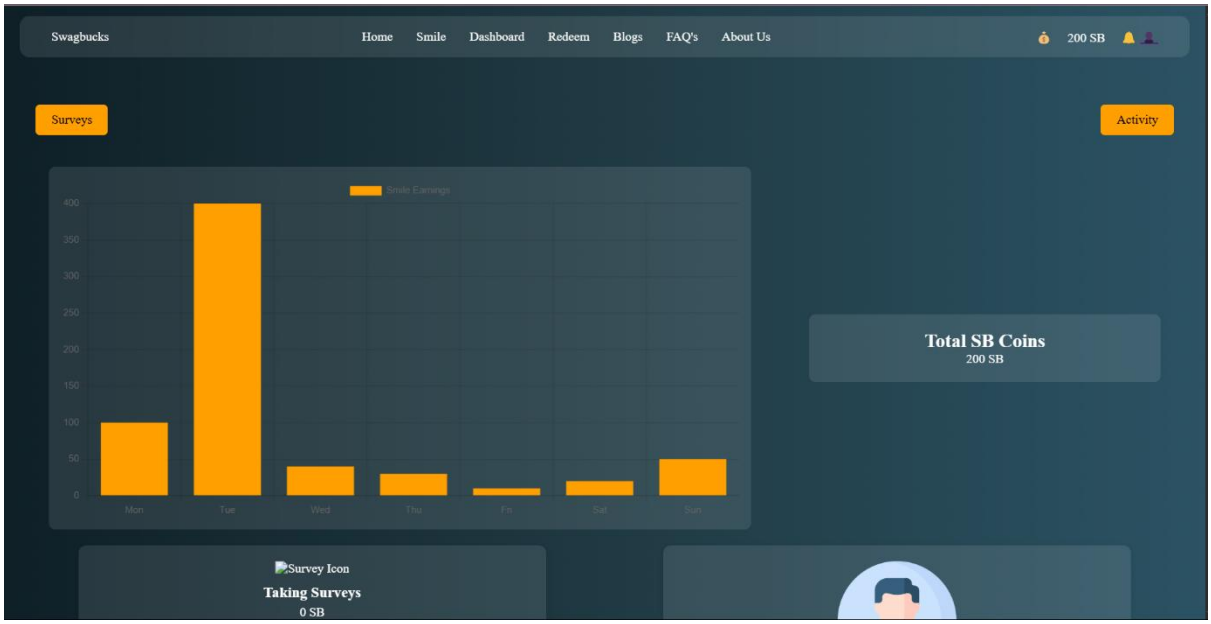


Figure 11.9 Dashboard

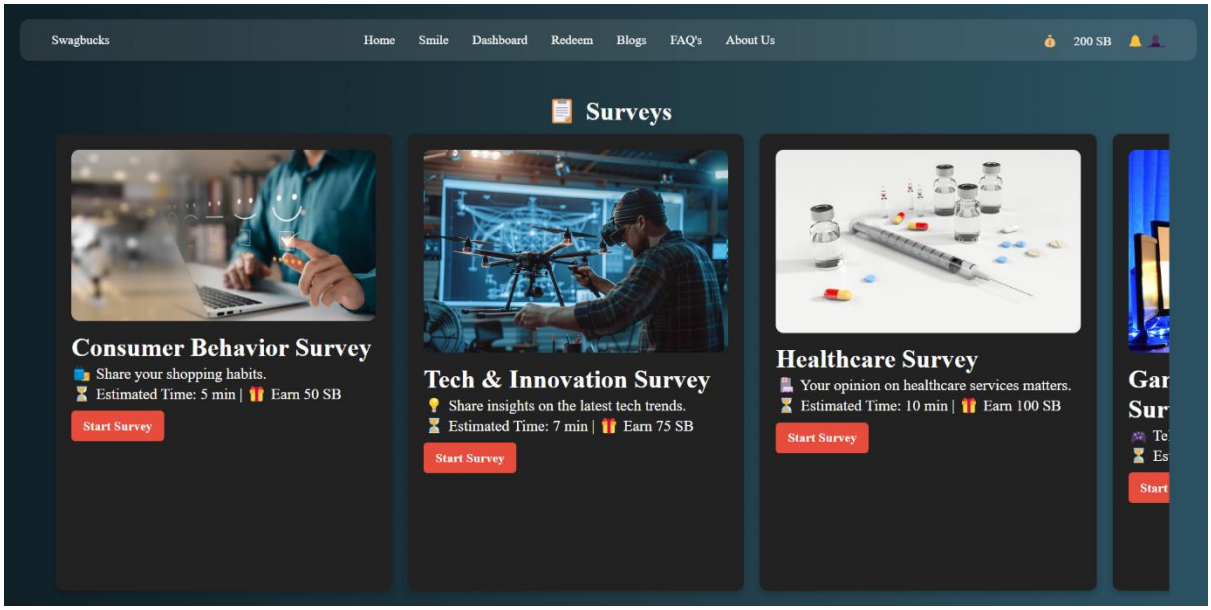


Figure 11.10 Survey Page

Figure 11.11 Redeem Page

Figure 11.12 Detailed Redeem Page

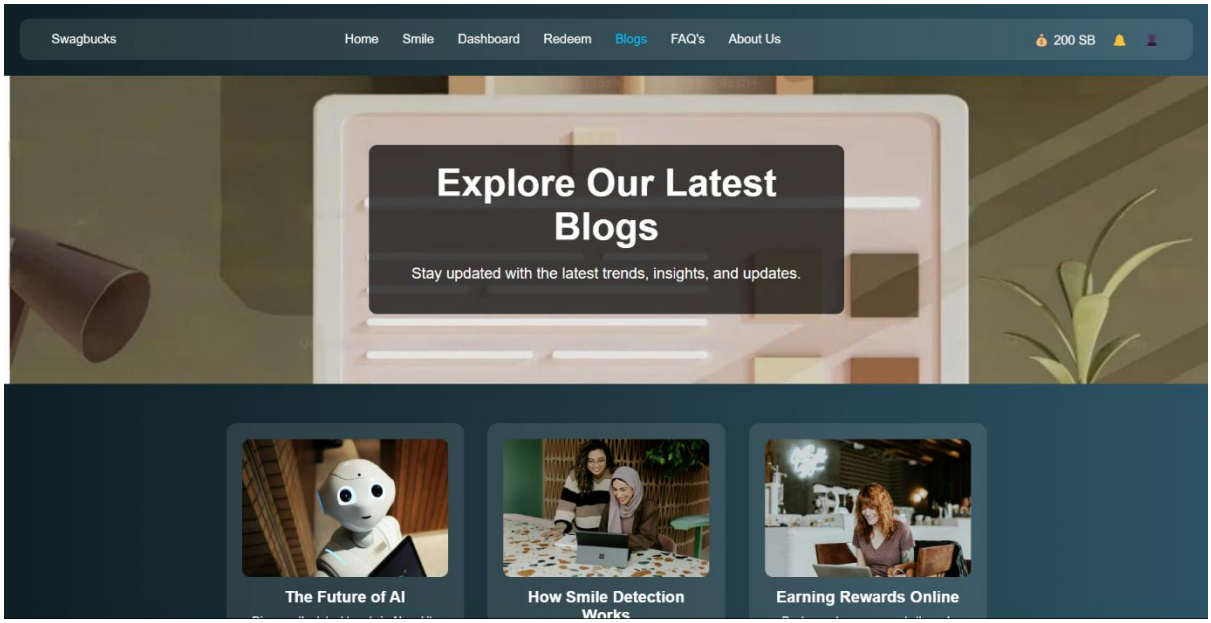


Figure 11.13 Blogs

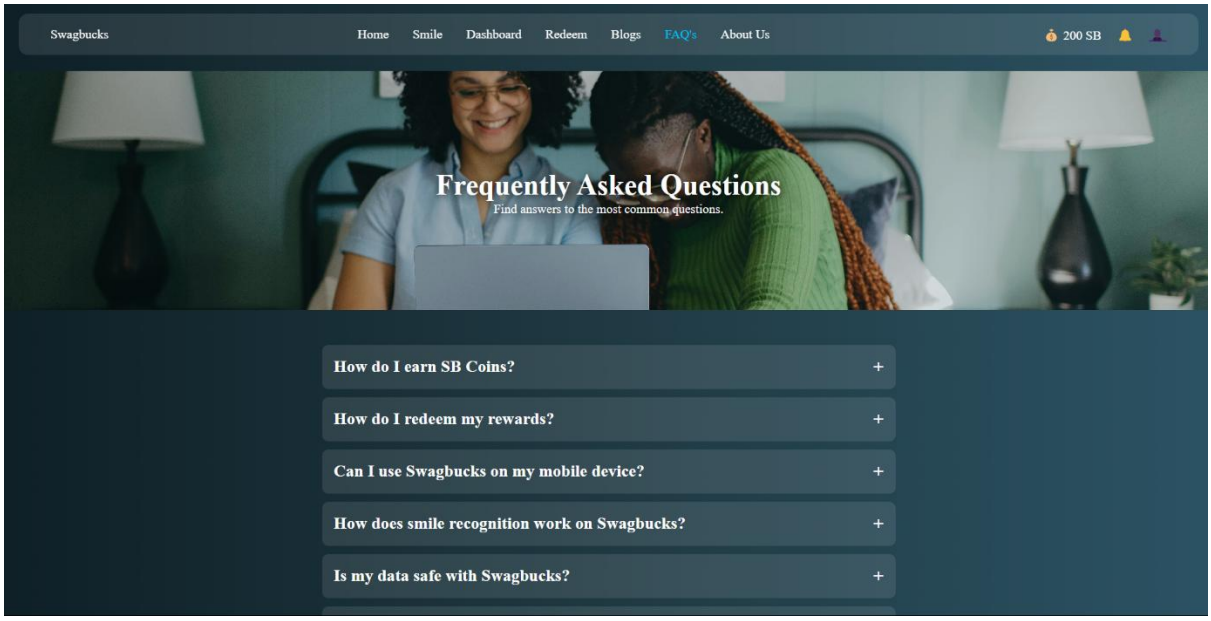


Figure 11.14 FAQ's

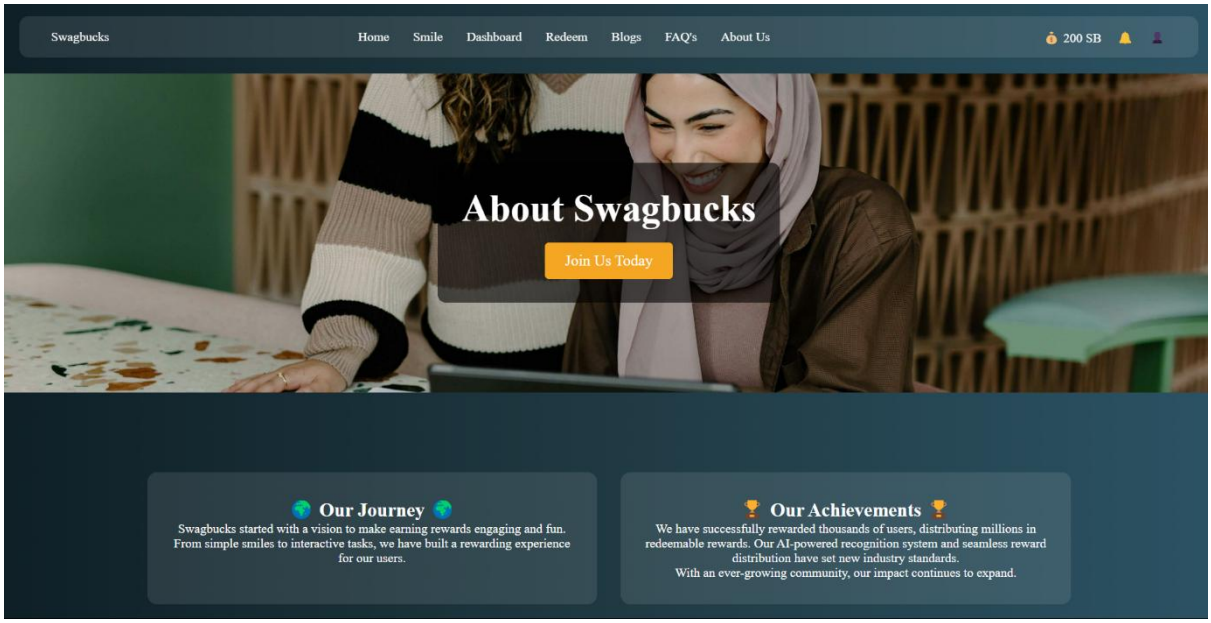


Figure 11.15 About Page

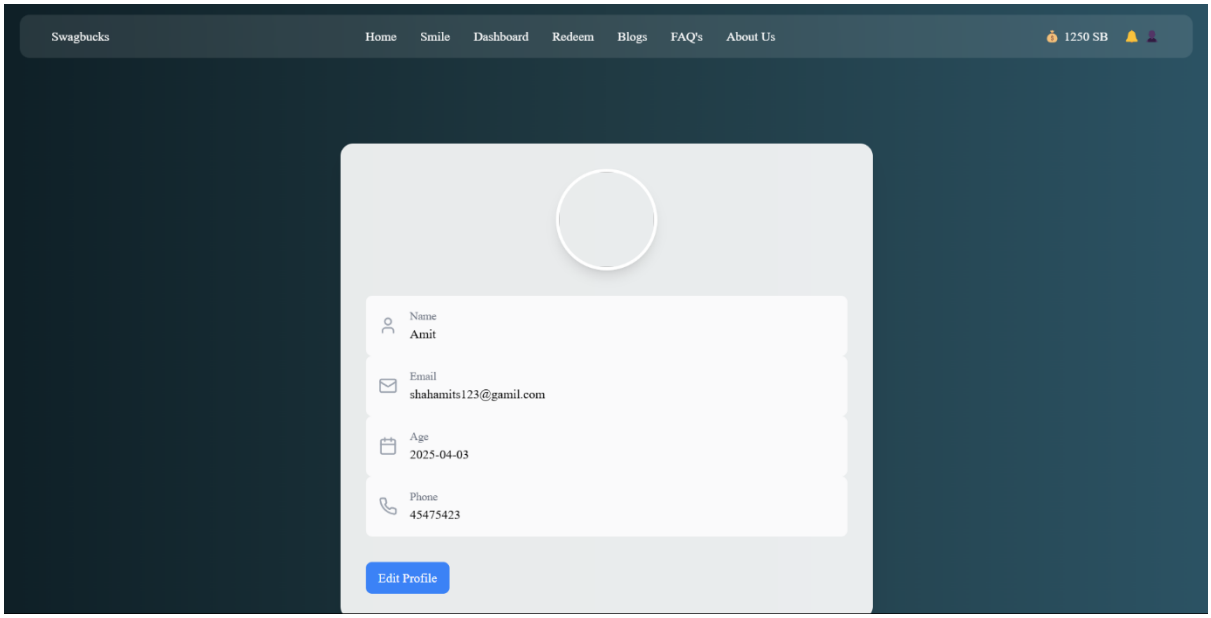


Figure 11.16 Account Page

e. Test Cases (Unit Test Cases & System Test Cases)

Test Case ID	Description	Input/Action	Output	Status
TC01	User Registration with valid details	Name, Email, Password	Success message, redirect login	✅
TC02	Smile Detection Accuracy	Smile in webcam feed	SB Coin credited	✅
TC03	Submit Survey	Selected answer	Coins added, next survey loaded	✅
TC04	Invalid login	Wrong email/password	Error message	✅
TC05	Referral works	Referred user signs up	Coins credited to referrer	✅
TC06	Admin Login	Admin credentials	Access to dashboard	✅

Table 11.1 Test Case

12. CODING

a. SQL commands for

i. database creation (along with constraints)

details

	_id ObjectId	Name String	Email String	Phone Int32	Age String	Password
1	ObjectId('67f68b1b91c2d68…	"Amit"	"shahamits123@gamil.com"	55	"2025-03-31"	555
2	ObjectId('67f68b71fd67e7c…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-02"	55
3	ObjectId('67f68b8badf7a61…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-02"	55
4	ObjectId('67f68bee538c7c8…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-02"	55
5	ObjectId('67f68d1b59fc486…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-01"	66
6	ObjectId('67f68e0e61b4681…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-02"	55
7	ObjectId('67f68e4c61b4681…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-01"	66
8	ObjectId('67f68e96845bb12…	"Amit"	"shahamits123@gamil.com"	777777	"2025-04-01"	77
9	ObjectId('67f68ee76f7995b…	"Amit"	"shahamits123@gamil.com"	77777	"2025-04-02"	77
10	ObjectId('67f68efd6f7995b…	"Amit"	"shahamits123@gamil.com"	45475423	"2025-04-02"	55
11	ObjectId('67f68f453297c02…	"Amit"	"shahamits123@gamil.com"	77777	"2025-04-02"	77

Table 12.1 Database Creation

ii. data insertion in tables

```
_id: ObjectId('67f68d1b59fc4868491a19fb')
Name : "Amit"
Email : "shahamits123@gamil.com"
Phone : 45475423
Age : "2025-04-01"
Password : 66
__v : 0
```

```
_id: ObjectId('67f68e0e61b468182732af90')
Name : "Amit"
Email : "murali@bala.com"
Phone : 457896631
Age : "2025-04-02"
Password : 2436
__v : 0
```

Table 12.2 Database Insertion

iii. access rights for different users.

In your project (Swagbucks-inspired reward platform), different types of users interact with the system. To maintain data privacy, system security, and functional integrity, access rights must be clearly defined and implemented.

Types of Users in the System:

1. Admin
2. Registered Users (Participants)

User Type	**Access Permissions**
Admin	○ Full access to all system modules ○ Can add/edit/delete surveys ○ View analytics and user activity ○ Manage user reports and referrals ○ Access all user data and feedback
User	○ Can register and log in securely ○ Can access and complete surveys ○ View their own dashboard and coin history ○ Refer friends ○ Upload smile selfies ○ No access to admin modules

Table 12.3 User Types

b. Comments & Description of Coding Segments

This section provides a detailed explanation of the major code modules, their functionality, and how they work together within the application. The backend is implemented using Node.js, Express, and MongoDB with Mongoose.

- **User Registration & Authentication Module**
  - **Purpose**: Allows users to register, login, and manage sessions securely.
  - **Code Description**:
    - Accepts user input (name, email, password).
    - Password is hashed using bcrypt for security.
    - After successful registration/login, a JWT token is generated and sent to the user.
    - Code ensures unique email verification and handles duplicate entries.
  - **Security**: All sensitive data is sanitized and stored securely.

- **Survey System Logic**
  - **Purpose**: Display available surveys to users and log their responses.
  - **Code Description**:
    - Retrieves surveys from the surveys collection.
    - When a user completes a survey, a log entry is created in coinsLog and the user's coin count is updated.
    - Survey completion is validated to avoid duplicate submissions.
    - Modular functions are used to manage survey creation (for admin), listing, and submission.

- **Referral System**
    - **Purpose**: Allow users to invite others and earn coins.
    - **Code Description**:
        - Generates a unique referral code per user.
        - When someone signs up using that code, the inviter earns bonus coins.
        - A coins log entry is created with source: "referral".

- **Dashboard Analytics**
    - **Purpose**: Provide real-time feedback on user activity.
    - **Code Description**:
        - Backend API endpoints fetch analytics like coin history, number of surveys completed, etc.
        - Data is structured and filtered using MongoDB's aggregate pipeline.

- **Smile Detection Module**
    - **Purpose**: Users earn coins for smiling using face detection.
    - **Code Description**:
        - Uses face-api.js to detect smiles through webcam.
        - Smile is verified based on confidence threshold.
        - Coins are awarded once per session to avoid spamming.

- **Error Handling and Response Structure**
    - All APIs return consistent JSON responses with status, message, and data.
    - Errors are caught using centralized error middleware.
    - Each controller function contains try-catch blocks to avoid server crashes.

- **Code Organization**
    - Code is organized using MVC pattern:
        - **Models**: MongoDB schemas via Mongoose.
        - **Controllers**: Business logic and database handling.
        - **Routes**: API endpoints for users, surveys, auth, etc.
    - Reusability and scalability are ensured via modular functions.

13. STANDARDIZATION OF THE CODING

a. Code Efficiency & Error Handling

Code Efficiency

- All modules follow DRY (Don't Repeat Yourself) principles, reusable functions are defined in utility files or service layers.
- Heavy operations like image processing or data filtering are optimized using asynchronous programming to avoid blocking the main thread.
- Database queries are written using indexes and pagination (if needed) to prevent performance bottlenecks.
- Smile detection runs on the client-side to offload server load.

Error Handling

- All APIs are wrapped in try-catch blocks to handle unexpected failures gracefully.
- User-friendly error messages are returned to the frontend, and technical errors are logged internally.

b. Parameters calling/passing

- Functions and API routes use clear, minimal, and descriptive parameters.
- Default values are set for optional parameters to prevent undefined errors.
- For long parameter lists, objects are used to improve readability.

c. Validation Checks

Both frontend and backend validations are implemented to ensure safe and accurate data flow.

Frontend Validations

- Required fields, email formats, and password rules are checked using form libraries or custom logic.
- File upload limits (image size, format) are validated before submission.

Backend Validations

- Inputs are sanitized to prevent SQL injection or XSS attacks.
- Smile scores are verified to be numeric and within valid thresholds.
- Duplicates are prevented using constraints and unique checks.

Additional Best Practices Used

- ESLint + Prettier used for code formatting.
- Modular directory structure for clarity (/components, /services, /api, /utils).
- Comments are added throughout for documentation.
- Git version control is used with meaningful commit messages.

14. TESTING

a. Testing Techniques & Testing strategies used

Techniques Used:

1. **Black Box Testing:**
    - Focused on user inputs and expected outputs without looking at internal code structure.
    - Error handling scenarios were tested, such as incorrect passwords, empty survey responses, and invalid image formats.
    - Used for: Login, Survey submissions, Smile uploads.

2. **White Box Testing:**
    - Involved checking the internal code logic, loops, and branches.
    - Unit tests were written for utility functions like smile score validators and referral matchers to isolate logic errors early.
    - Used in backend API testing and reward calculation logic.

3. **Regression Testing:**
    - After each major update (e.g., new reward system), previously working features were re-tested.
    - Automated regression scripts checked login flow, coin balance updates, and survey response storage.

4. **Usability Testing:**
    - Non-technical users were asked to use the app and report any difficulties or confusion.
    - Feedback led to UI improvements such as adding hover tooltips, simplifying coin summary layout, and increasing contrast for readability.

b. Testing Plan

We used a structured V-Model Testing Plan, where every development stage was followed by a corresponding testing phase:

Development Stage	Corresponding Test
Requirement Gathering	Acceptance Testing
System Design	System Testing
Module Design	Integration Testing
Coding	Unit Testing

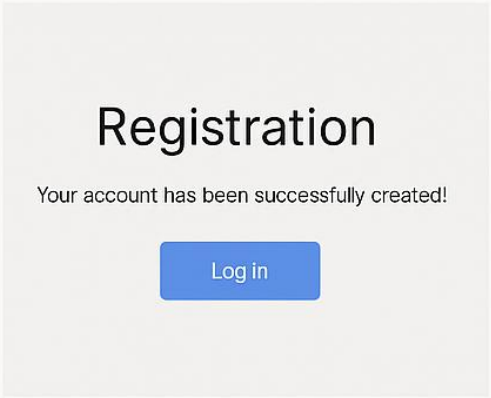
Table 14.1 Testing Plan

Testing Environments:

- Local Development Server for backend API testing.
- Browser Testing for responsive UI (Chrome, Firefox, Edge, Mobile).
- Firebase Emulator for testing database/auth without production risk.

c. Test Reports for Unit Test & System Test Cases

Unit Test Case	
TC_U001	User Registration
Module	Register with valid email & password
Input Data	Email: user@test.com Password: Test 123
Expected Result	Account created, redirect to login page
Actual Result	Successfully registered and redirect



System Test Case	
TC_S005	Admin Analytics
Input	Admin views charts
Expected Output	Graphs of total users, coins distributed, engagement
Actual Result	Admin panel displayed with all stats

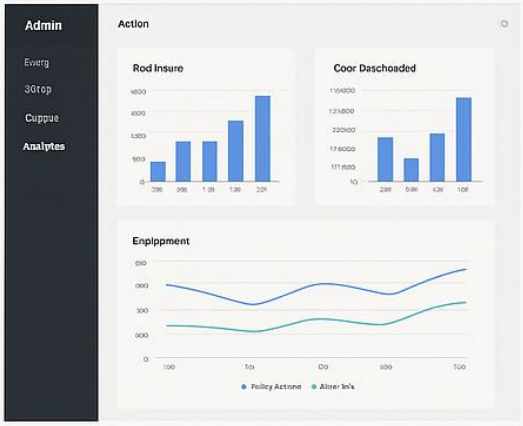


Figure 14.1 Test Report

d. Debugging & Code Improvement

Throughout development, we encountered issues that were fixed using:

Debugging Tools:

- **Chrome DevTools**: Used for testing UI behaviour and API calls.
- **Postman**: For testing backend endpoints independently.
- **Console Logs + Firebase Logs**: Helped trace errors in cloud functions and backend services.

Improvements Made:

- Replaced large image uploads with compression to improve speed.
- Optimized smile detection to only process once per second, reducing CPU usage.
- Added retry mechanism in API calls to avoid network-related failures.
- Enhanced error feedback in UI for better user understanding.

15. SYSTEM SECURITY MEASURES

a. Database/ Data security

The system uses robust data security practices to protect user information and application data:

- **Hashed Passwords**: All user passwords are hashed using strong cryptographic functions like bcrypt, ensuring no raw credentials are stored in the database.
- **Parameterized Queries**: To prevent SQL Injection, all database operations use prepared statements or ORM-based methods.
- **Encrypted Communication**: All client-server interactions are conducted over HTTPS, using SSL/TLS encryption, protecting data in transit.
- **Data Backups**: Regular, automated backups are scheduled to prevent data loss in case of any failure or breach.
- **Restricted Database Access**: The production database is only accessible to the application server, and not exposed to the public internet. Admin credentials are secured using environment variables

b. Creation of User Profiles and access rights

To manage users securely and offer role-based access control, the platform includes:

- **Authentication System**: Users register and log in using their email and password. Sessions are maintained using JWT (JSON Web Tokens) to secure API access.
- **User Roles**:
    - **Normal Users**: Can access features like surveys, selfie uploads, dashboards, and coin redemption.
    - **Admins**: Have additional privileges to view analytics, manage user data, monitor usage trends, and handle reported issues.
- **Role-Based Routing**: The backend includes logic that checks the user's role and grants or denies access to certain API endpoints accordingly.
- **Access Control**: Sensitive operations (e.g., editing rewards, viewing all user activity) are restricted to admin-level users only.
- **Session Expiry & Logout**: JWT tokens are time-limited and expire after a set duration to prevent long-term unauthorized access.

16. COST ESTIMATION OF THE PROJECT

Cost estimation is a critical step in the planning and execution of any software project. It helps in predicting the effort, resources, and budget required to complete the project successfully. For our rewards-based web platform, we have used the **COCOMO (Constructive Cost Model)** as the cost estimation model.

COCOMO Basic Model Applied

The COCOMO model calculates cost based on the size of the project measured in **KLOC (Kilo Lines of Code)** and other project factors.

Assumptions:

- Project Type: **Organic**
- Estimated Code Size: ~5 KLOC
- Effort Estimation Formula: Effort (E) = a × (KLOC)^b
- For Organic projects, constants are: a = 2.4, b = 1.05
-

E = 2.4 × (5)^1.05 ≈ 13.5 Person-Months

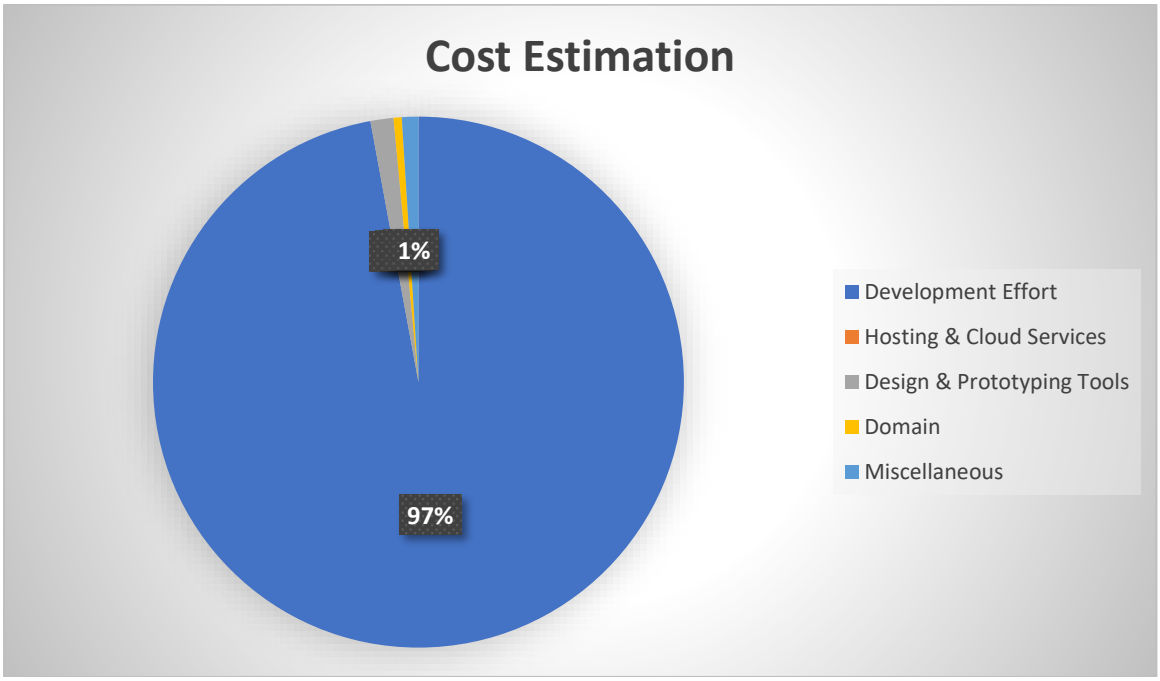


Figure 16.1 Cost Estimation

Estimated Cost Breakdown

Cost Component	Estimated Cost (INR)
Development Effort	₹2,02,500
Hosting & Cloud Services	₹0
Design & Prototyping Tools	₹3,000
Domain	₹1,000
Miscellaneous (Testing, Reports, Internet)	₹2,000
Total Estimated Cost	**₹2,08,500/-**

Table 16.1 Cost Breakdown

17. REPORTS

This section showcases how data is managed and reported to users and admins through your website. It includes sample report layouts, such as what users or administrators can see related to earnings, activity, survey performance, etc.

Types of Reports:

1. User Activity Report

- Shows total SB coins earned by a user.
- Displays a breakdown of coins from:
    - Surveys
    - Smile/selfie activity
    - Referrals
- Time range filters (Today / Last 7 days / All-time)

2. Admin Dashboard Report

- Summary of platform performance:
    - Total active users
    - Most completed surveys
    - Referral growth trends
    - Top coin earners
- Insight into what interest's users the most

3. Survey Performance Report

- For each survey:
  - Number of completions
  - Average rating (if applicable)
  - User demographics (optional)

4. Smile Detection Logs

- Track how many users interacted with smile-based coin earning.
- Show timestamps of selfie submissions.

18. FUTURE SCOPE

The development of the Swagbucks rewards platform marks a significant step toward a more engaging, gamified web experience. However, there is ample room for growth and evolution. As the user base expands and technology advances, the following future enhancements are proposed:

1. **AI-Based Personalization**

   Implement AI/ML models to analyze user behaviour and recommend surveys, offers, and rewards that match their interests. This will improve engagement and increase participation rates.

2. **Mobile Application (Android & iOS)**

   Creating a cross-platform mobile app will greatly enhance accessibility and allow users to engage with the platform anytime, anywhere. Push notifications for survey availability and reward reminders can also be added.

3. **Advanced Analytics Dashboard**

   Upgrade the admin panel to include deeper data visualizations like funnel analysis, cohort analysis, and real-time traffic heatmaps. This will empower admins to make data-driven decisions.

4. **Blockchain-Based Rewards System**

   Integrate blockchain for transparent and secure reward transactions. Users can redeem coins as tokens which can potentially be traded or used on other platforms in the ecosystem.

5. **Gamification Upgrades**

   Introduce features like:

   - Streak rewards for daily activity
   - Leaderboards for top contributors
   - Badges and achievements

   This will help maintain user motivation and retention over time.

6. **API Integration with Third-Party Platforms**

   Allow businesses to create and post their own surveys directly on the platform through a developer API. This opens up monetization opportunities and partnerships.

7. **Multi-Language Support**

Expand user accessibility by offering the platform in multiple languages, especially regional languages, to cater to a diverse demographic across different geographies.

8. **Enhanced Security Measures**
    - Two-factor authentication (2FA).
    - Facial recognition-based selfie validation.
    - Encrypted storage of user media and data.

This ensures safety and builds user trust.

9. **Real-Time Moderation Tools**

Develop a real-time content moderation system for uploaded images or comments using AI-powered filters, reducing the admin overhead and improving content quality.

19. BIBLIOGRAPHY

This section outlines all the reference materials, tools, documentation, and learning resources that contributed to the successful development of the Swagbucks rewards platform. These sources were essential in shaping the technical backbone of the application, guiding design principles, ensuring secure development practices, and aligning the platform with industry standards.

Throughout the project, various online tutorials, official documentation, research articles, and textbooks were consulted to gain a deeper understanding of topics like serverless architecture, facial recognition using face-api.js, user interface best practices, cloud deployment, and full-stack web development. These resources not only helped in resolving technical challenges but also enriched the overall quality and efficiency of the implementation process.

By acknowledging these references, we ensure academic integrity and express gratitude to the developers, authors, and communities whose work and insights have significantly influenced our project's direction and success.

Books & Research Papers

1. *Web Development with Node and Express* by Ethan Brown
2. *Designing with Web Standards* by Jeffrey Zeldman
3. *Don't Make Me Think* by Steve Krug – For UI/UX Principles
4. *Clean Code: A Handbook of Agile Software Craftsmanship* by Robert C. Martin
5. *Agile Estimating and Planning* by Mike Cohn

Web Resources

1. MDN Web Docs – HTML, CSS, and JavaScript Reference
2. W3Schools – Web Development Tutorials
3. Face-api.js Documentation
4. ReactJS Official Docs
5. Node.js Official Docs
6. AWS Documentation – Serverless Architecture
7. MongoDB Documentation
8. Firebase Authentication Docs

Software & Tools Used

- draw.io – For system diagrams and modelling
- Canva – For UI wireframes and visuals
- Figma – For interface design prototyping
- GitHub – Version control and code hosting
- Visual Studio Code – Code editor for development
- MongoDB – Backend & Database

20. APPENDICES

The appendix section includes all the supplementary materials that support the development, design, and evaluation of the Swagbucks-inspired rewards platform but are not part of the main body of the report. These materials provide additional clarity, evidence, and insight into the technical and functional aspects of the system.

Appendix A – Survey & Smile Detection Screenshots

Screenshots showing how users interact with surveys and the AI-powered smile detection module. These include:

- Live smile tracking via webcam
- Survey submission pages
- Reward confirmation pop-ups

Appendix B – Database Schema

A complete visual representation of the MongoDB schema used in the project, including collections such as:

- Users
- Surveys
- SmileLogs
- Rewards
- Referrals

Appendix C – Sample Code Snippets

Highlighted snippets showing key features such as:

- Face detection logic with face-api.js
- Serverless function for storing coins
- Authentication using Firebase

- Reward redemption flow

Appendix D – Gantt Chart and PERT Chart

Planning diagrams showing the timeline and dependencies of various project phases:

- Requirement gathering
- Design
- Development
- Testing
- Deployment

Appendix E – Test Case Evidence

Annotated screenshots and reports that validate each test case execution, including:

- Unit test output
- UI validation tests
- Survey flow functionality tests
- Error handling and failure recovery logs

Appendix F – Sample Survey and Referral Report Formats

PDF or screenshot samples of reports generated by the admin panel, showing:

- User activity logs
- Coin earnings report
- Survey participation breakdown
- Referral engagement stats

21. GLOSSARY

This glossary provides comprehensive definitions and explanations for the technical terms, tools, and concepts mentioned in the report. It is designed to help readers of all backgrounds understand the various technologies, frameworks, and methodologies applied in the development of the Swagbucks-inspired rewards platform.

Swagbucks

Swagbucks is an online rewards platform that incentivizes users for completing micro-tasks such as answering surveys, watching videos, shopping online, and referring friends. The concept has inspired our platform, which implements similar earning features through innovative additions like smile detection and gamification.

SB Coins

SB Coins are the virtual currency or reward points that users earn on the platform for participating in activities like completing surveys, smiling in front of the camera, or referring others. These coins accumulate in the user's wallet and can be redeemed for real-world benefits such as gift cards, vouchers, or merchandise.

Smile Detection

Smile Detection is a feature integrated using facial recognition technology that identifies when a user is smiling through their webcam. On successful detection, the system automatically rewards the user with SB Coins. This feature enhances engagement through gamified interaction.

Surveys

Surveys are structured questionnaires presented to users to gather their opinions, preferences, or data on specific topics. On submission, users are rewarded with SB Coins. These surveys may serve academic, commercial, or market research purposes.

face-api.js

A powerful JavaScript library built on TensorFlow.js for performing real-time face detection and facial expression recognition directly in the browser. It was used to implement the smile detection system in this project, offering a client-side AI solution without requiring external APIs or server-side processing.

Serverless Architecture

Serverless architecture refers to a model where cloud providers automatically manage the infrastructure and resource provisioning needed to run applications. Developers only need to focus on writing the application logic. In this project, AWS Lambda and similar serverless tools were considered for backend processes, ensuring scalability and low maintenance.

MongoDB

A NoSQL database designed for high performance and scalability. It stores data in flexible, JSON-like documents. MongoDB was used to store user profiles, smile logs, survey results, and referral histories. Its document-based nature made it ideal for handling structured yet flexible user data.

Node.js

An open-source, cross-platform JavaScript runtime environment that executes code outside a web browser. It is commonly used to build scalable and efficient server-side applications. Node.js powered the backend APIs in this platform.

Data Flow Diagram (DFD)

A graphical representation used to visualize how data flows through a system. It shows the movement of information between processes, data stores, and external entities. DFDs were used in this project to model system workflows and backend interactions.

PERT Chart (Program Evaluation Review Technique)

A project management tool used to schedule, organize, and coordinate tasks. It helps estimate the minimum time required to complete the project by identifying dependencies and critical paths.

Gantt Chart

A horizontal bar chart used to illustrate a project schedule. It outlines the start and end dates of each development phase, milestones, and dependencies. This chart helped track progress and align efforts with deadlines.

Authentication

The process of verifying the identity of users before granting access to system resources. Firebase Authentication was implemented to secure login and user-specific functionality in the platform.

Authorization

Determines what resources a verified user can access or modify. Based on user roles (admin/user), access rights to surveys, dashboards, and reports were controlled in the system.

SRS (Software Requirements Specification)

A formal document that defines the software's intended features, behavior, and constraints. It served as a blueprint for developers and stakeholders during this project.

Unit Testing

A software testing method where individual components or functions are tested independently to ensure they work as expected.

System Testing

A high-level test where the entire system is tested as a whole to validate that it meets the specified requirements and performs well in real-life scenarios.

******