EDUDONATE

CODE CRUSHERS:

Kavyadharshini M
Akshaya Sreekumar
K Madhushree

INTRODUCTION:

We are developing a simple C++ app that helps track study time and rewards users for their dedication. The app records how many hours a person studies each week and gives points based on the time spent.The goal is to reach 5000 points in a year, at which a donation of 50 rupees is made by our sponsors. After reaching the target, points reset to zero, and the cycle starts again. Users who contribute 100 rupees within the year will receive a gift voucher.

The app is designed to be user-friendly, with features like login, points tracking, badges, and a Hall of Fame. Users can also collaborate with friends and see how much money has been donated. Badges are awarded based on study hours, and the Hall of Fame shows who has earned the most badges.

This app not only encourages consistent studying but also supports charitable causes, making it a fun and meaningful tool for students.

PROBLEM STATEMENT:

In today's fast-paced world, students often struggle to maintain consistent study habits, which can negatively impact their academic performance and personal growth. Additionally, there is a growing need for platforms that encourage social responsibility and community engagement among young people. However, most existing study apps focus solely on productivity tracking without offering incentives or fostering a sense of community and contribution to society.

SOLUTION:

This project aims to address these issues by developing a user-friendly app that not only tracks study time but also rewards users for their dedication. By integrating a point-based system, badges, and the opportunity to contribute to charitable causes, this app motivates students to stay consistent in their studies while also giving back to the community. It bridges the gap between personal development and social responsibility, making the learning process both productive and impactful.

Furthermore, the app builds a sense of community by enabling users to collaborate with friends, compare progress, and engage in friendly competition. The Hall of Fame showcases top achievers, promoting healthy competition and sustained motivation. This approach not only improves study habits but also fosters a culture of learning and giving, encouraging a more engaged and socially responsible student community.

FEATURES:

1.Study Time Tracking

2.Point System

3.Badge System

4.Hall of Fame

5.Contribution Page

6.Daily Streaks

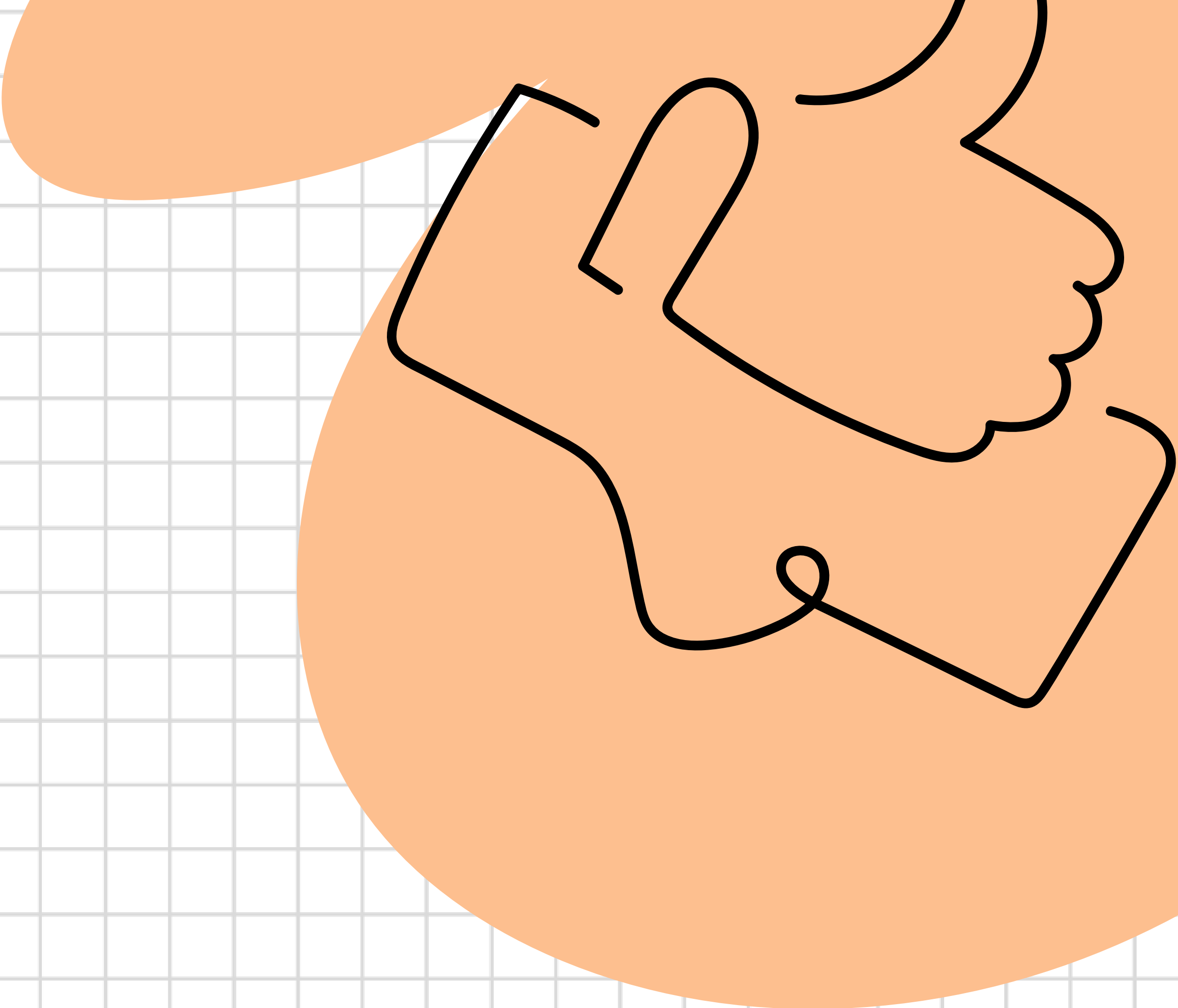
7.Challenges and Rewards

8.Data Security

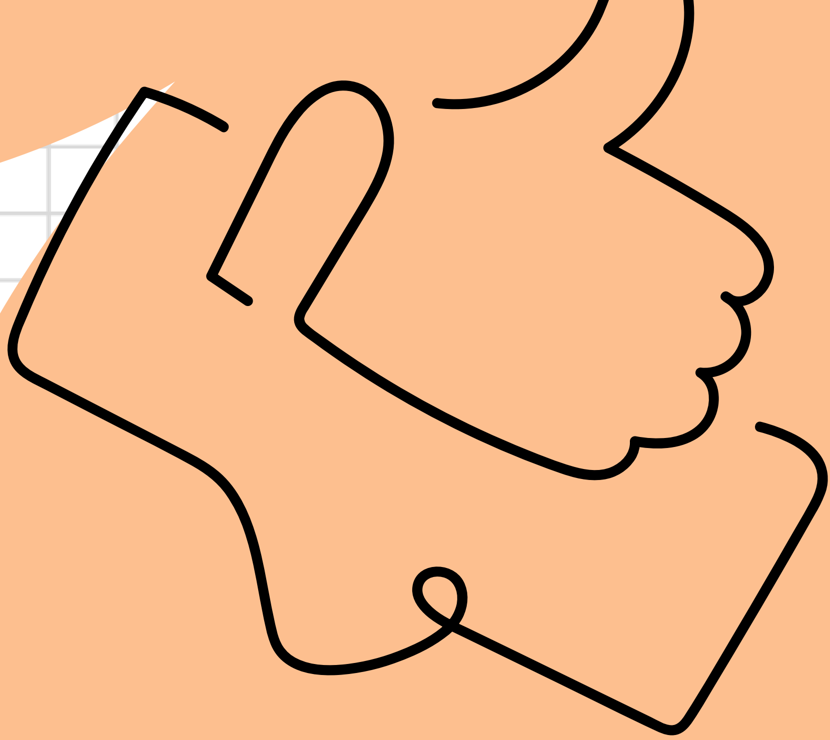
9.Progress Reports

10.Social Integration

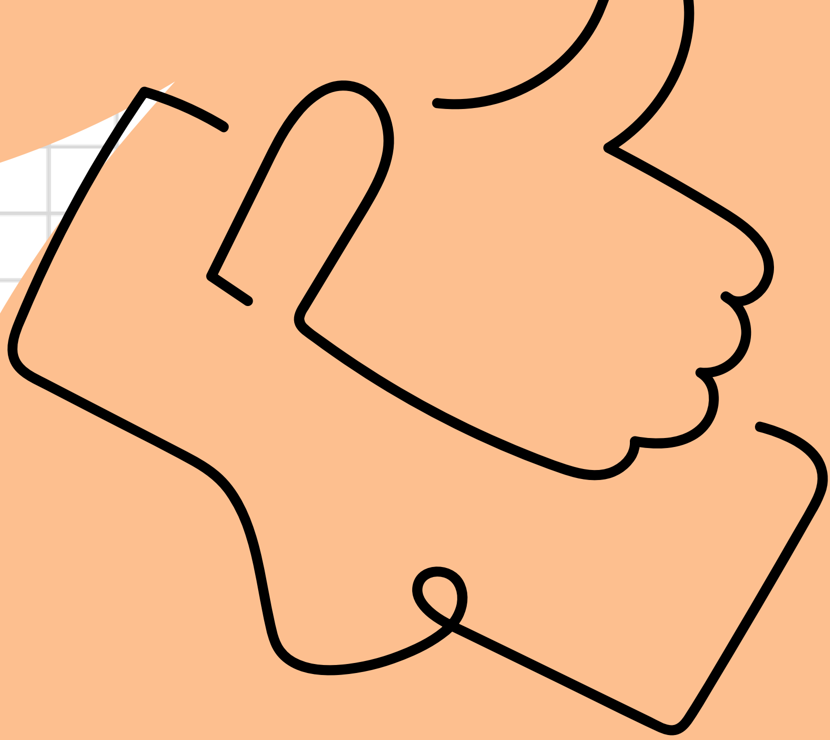
11.Premium Features (Optional)



STUDY TIME TRACKING:



1. Data Structures: Use a class to manage the study sessions and another class to manage the overall tracking and point calculation.
2. Logging Study Sessions: Implement functions to log the start and end times of study sessions and calculate their duration.
3. Weekly Calculation: At the end of each week, calculate the total study hours and award points based on predefined thresholds.
4. Reset Mechanism: Reset the weekly study hours and accumulate the points for long-term tracking.



POINTS SYSTEM:

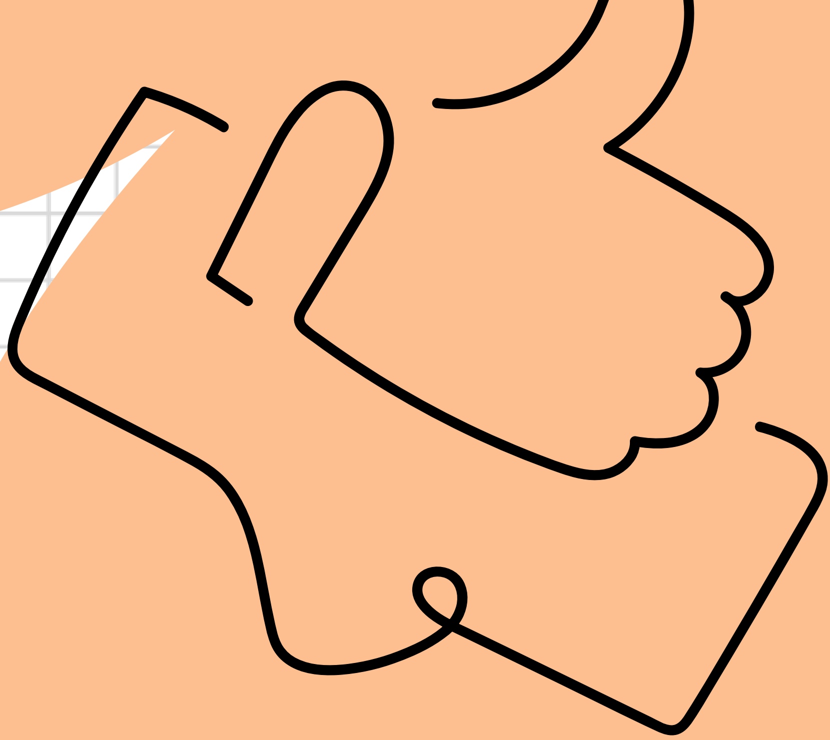
1. Logging Study Hours: The system will track the start and end times of study sessions and calculate the total duration.
2. Weekly Aggregation: The system will aggregate the study hours for each week.
3. Points Calculation: Points will be calculated based on the total study hours in a week according to predefined thresholds.
4. Display and Reset: The system will display the points earned and reset the weekly hours at the start of a new week.

BADGE SYSTEM:

We'll design a system that awards badges based on the total number of study hours logged in a week using structures. The badge thresholds will be:

- Bronze Badge: 10-12 hours per week
- Silver Badge: 12-14 hours per week
- Gold Badge: 14-20 hours per week
- Diamond Badge: More than 20 hours per week

HALL-OF FAME: [EXTRA FEATURE]



Here’s a simple way to implement a Hall of Fame feature in C++:

1. Define User Profiles
   - User Info: Each user has a name, total study hours, and badges.
   - Data Structure: Use a class or struct to store this information.
2. Manage User List
   - Store Users: Keep a list of user profiles. This can be done using an array or a vector.
   - Add Users: Add new user profiles to the list as needed.
3. Rank Users
   - Sorting: Sort the list of users based on their total study hours or badges.
   - Criteria: Decide whether to sort by hours or badges to determine the top performers.
4. Display Hall of Fame
   - Show Top Users: Display the top users from the sorted list.
   - Details: Include details like name, hours, and badges.
5. Update and View
   - Update: After a study session, update the user’s profile and re-sort the list if needed.
   - View: Provide an option to view the Hall of Fame in the application menu.

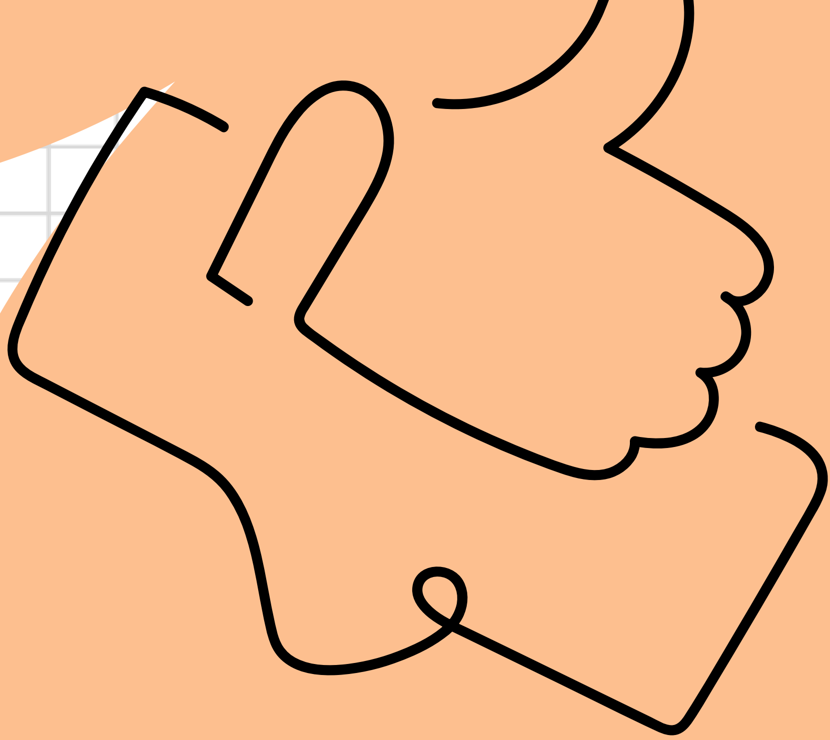
CONTRIBUTION PAGE:

1. Define Contribution Records
   - Attributes: Each contribution record should include attributes like the contributor's name, the amount contributed .

2. Display Contributions
   - List of Contributions: Provide a way to view all contributions made. This could include the contributor’s name, the amount contributed, and the date of the contribution.

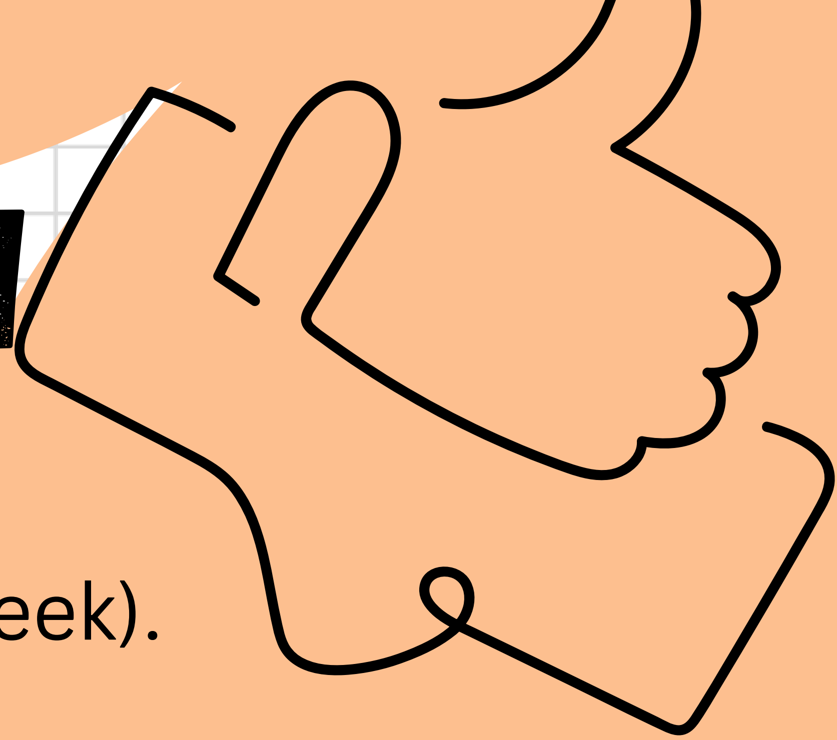
3. Security and Validation
   - Input Validation: Ensure that the amount contributed is a positive number and validate other inputs as necessary.

DAILY STREAKS:



1. Track Key Data
   - User Information: Keep track of:
     - Last Study Date: The date when the user last studied.
     - Current Streak: Number of consecutive days studied.
2. Update Streak
   - When User Studies:
     - Update the Last Study Date to today’s date.
     - Check if the Last Study Date was yesterday:
       - If yes, increase the Current Streak by 1.
       - If not, reset the Current Streak to 1.
3. Display Streak
   - Show Streak Info: Let users see their Current Streak whenever they check their profile.

CHALLENGES AND REWARDS:[EXTRA FEATURE]



1.User Starts a Challenge:

- Define and track the challenge criteria (e.g., study 2 hours daily for a week).

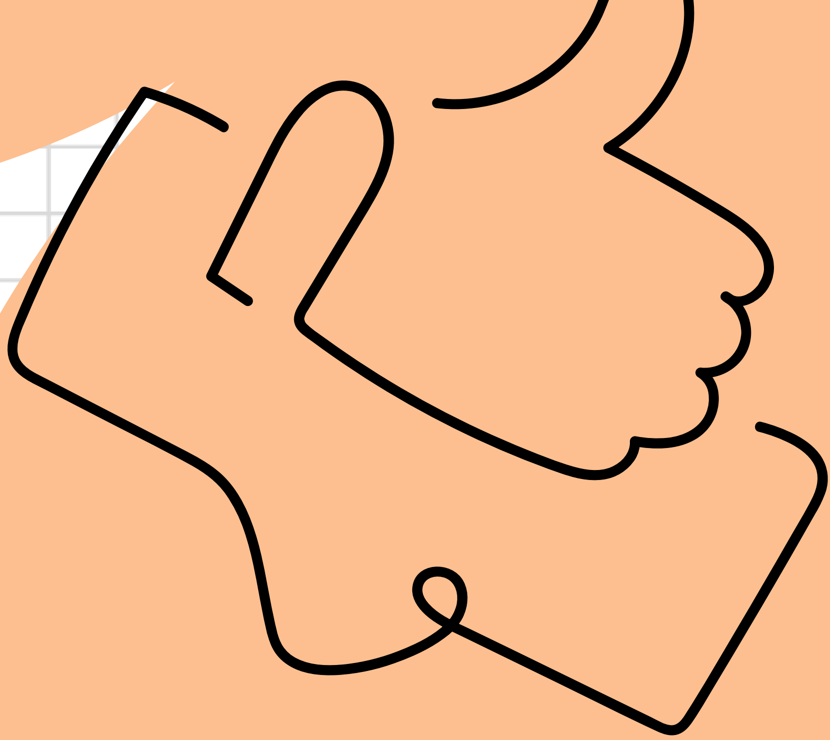
2.User Meets Criteria:

- Update their profile with the appropriate rewards (e.g., points, badges).

3.Display Achievements:

- Show the completed challenges and earned rewards on the user’s profile.

DATA SECURITY:

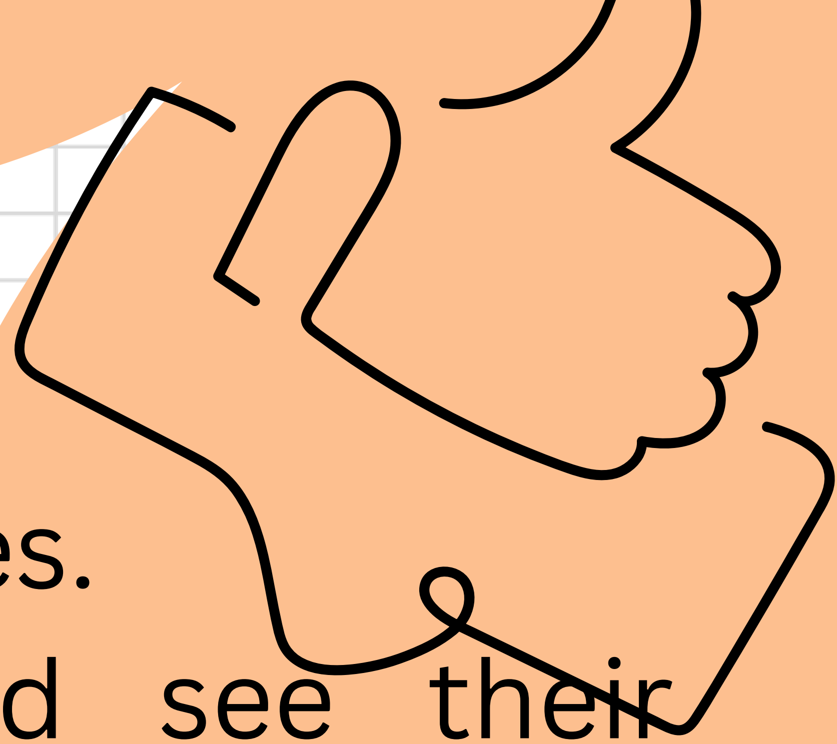


1. Control Access
   - Limit Access: Only allow authorized users to access sensitive data.
   - Authentication: Require users to log in.
   - Authorization: Set up roles to control what users can do.
2. Validate Input
   - Check Data: Ensure all user input is safe and expected.
3. Handle Errors Securely
   - Safe Error Messages: Don’t show sensitive information in error messages.
4. Update Regularly
   - Apply Patches: Keep software and libraries up-to-date to fix security issues.
5. Backup Data
   - Create Backups: Regularly back up important data and have a plan for recovery.
6. Follow Secure Coding Practices
   - Review Code: Regularly check your code for security flaws.

PROGRESS REPORT:

- Collect User Data: Track study hours, points, and badges.
- Generate Summary: Create a simple summary of the collected data.
- Display Summary: Show the report on the user’s profile page.

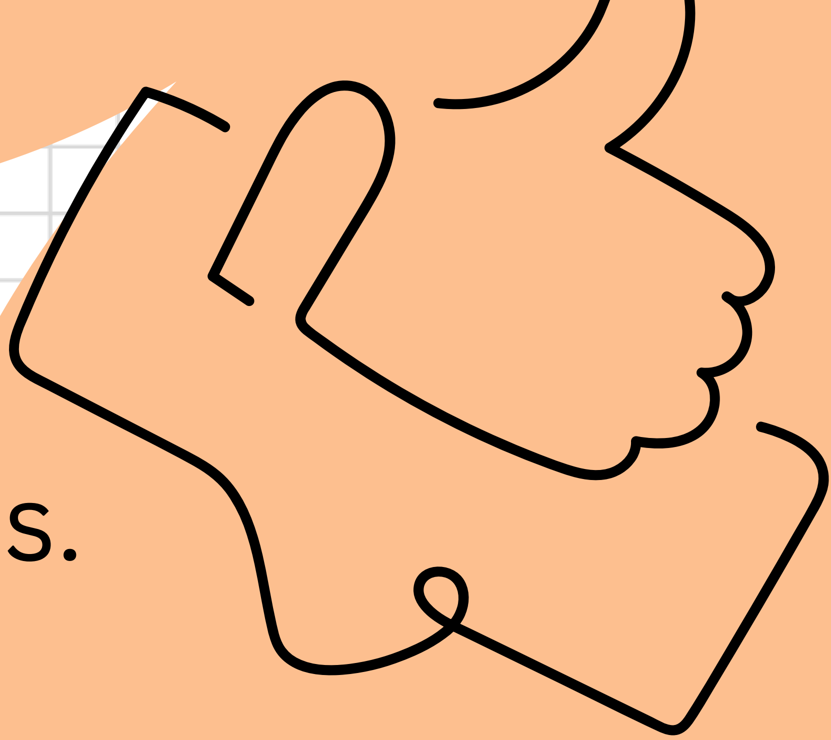
SOCIAL INTEGRATION:

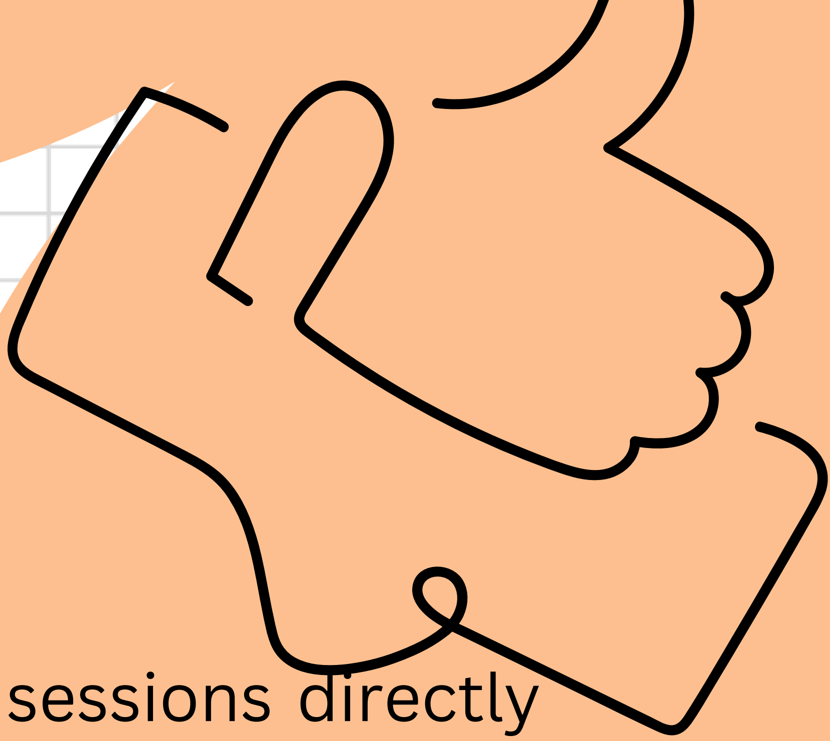


1. Profile Display: Users can view and edit their profiles.
2. Friend Interaction: Users can add friends and see their achievements.[FUTURE FEATURE]
3. Sharing Achievements: Users can share their progress on social media.[FUTURE DEVELOPMENT]
4. Community Engagement: Users can participate in forums and groups.
   [FUTURE FEATURE]

PREMIUM FEATURES:[EXTRA FEATURES]

1. Offers advanced analytics and personalized study plans.
2. Provides an ad-free experience for premium users.





USER-FRIENDLY INTERFACE:

1.Home Section

- Greeting: A friendly welcome message.
- Summary Stats: Quick stats such as total points and badges earned.
- Study Timer: A simple timer that users can start, pause, and reset to track their study sessions directly from the Home section. The time spent is automatically logged and contributes to the summary stats.

2. Profile Section

- User Details: Basic information and study goals.
- Edit Profile: Option to update personal details.

3. Progress Section

- Study Time: Simple display of study hours and progress.
- Points & Badges: List of points earned and badges.

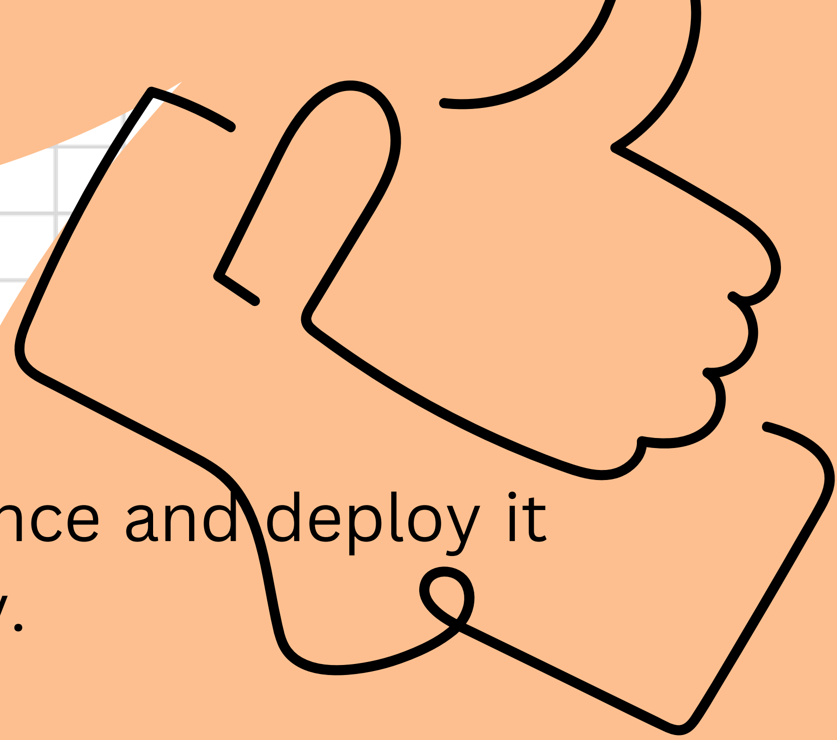
4. Challenges and Rewards

- Current Challenges: Overview of ongoing challenges.
- Rewards: Information on available rewards.

5. Help Section

- FAQ: Basic frequently asked questions.
- Support Contact: How to get help or contact support.

TECHNICAL APPROACH:



Framework Selection:

- Qt for Mobile: We use Qt because it’s cross-platform, meaning you can write the app once and deploy it on both Android and iOS. Qt also provides tools to create touch-friendly interfaces easily.

UI Design:

- Adaptive Layouts: Design the app to adjust automatically to different screen sizes and orientations. This makes the app look good on both phones and tablets.
- Basic Navigation: Implement simple navigation with buttons or tabs to switch between different sections of the app, like the dashboard, settings, and study logs.

Backend Integration:

- Local Data Storage: Store user data (like study hours) locally using a lightweight database .

Networking:

- If the app needs to sync data with a server, use simple HTTP requests to send and receive data.

Performance Considerations:

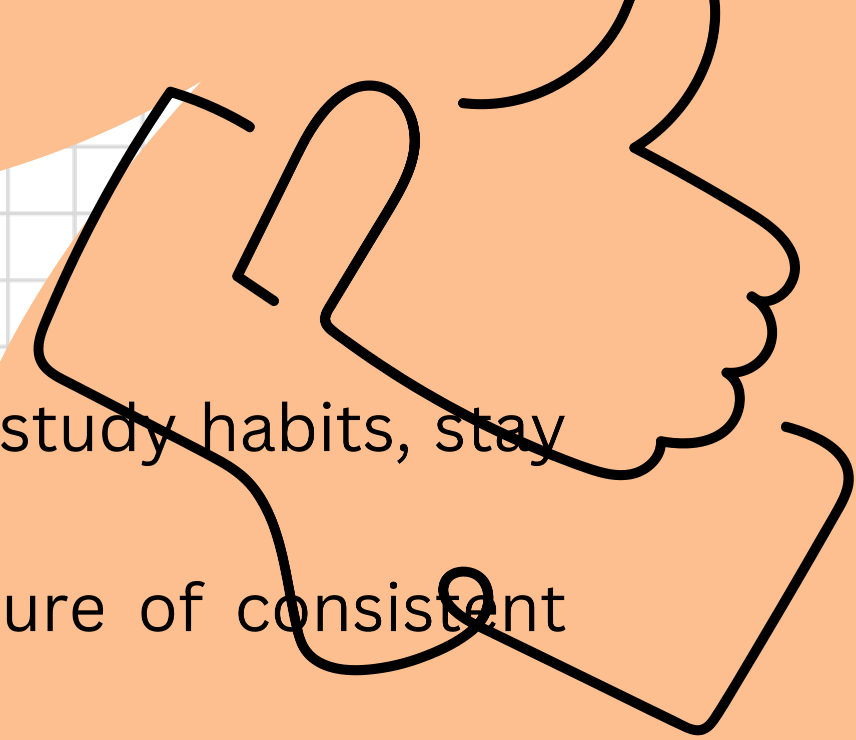
- Optimize for Mobile: Ensure the app runs smoothly by minimizing memory use and ensuring quick response times, particularly when dealing with touch inputs.

Testing:

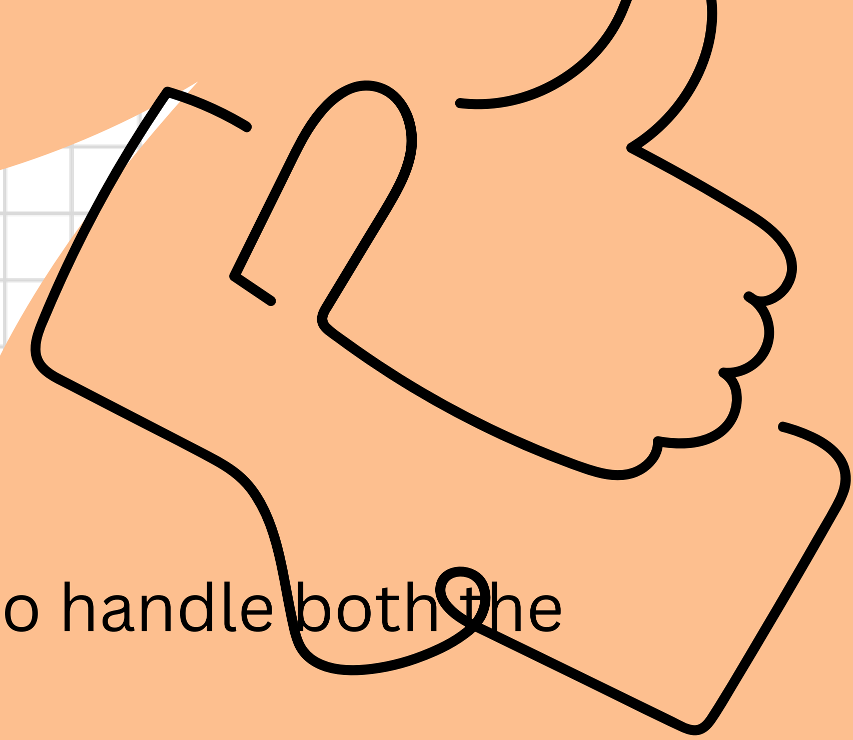
- Cross-Platform Testing: Regularly test the app on both Android and iOS devices to make sure it works well across different phones.

TARGET AUDIENCE:

- Students: Primarily aimed at high school and college students who want to track their study habits, stay motivated, and earn rewards for their dedication.
- Educational Institutions: Schools and colleges can use this app to encourage a culture of consistent studying and reward students for their efforts.
- Sponsors: Companies and organizations interested in supporting educational initiatives can use the app for branding and contributing to student development.



FEASIBILITY AND SCALABILITY:



FEASIBILITY:

C++ Libraries and Tools:

- The project is feasible with the selected C++ libraries and tools. C++ is powerful enough to handle both the backend logic and user interface requirements efficiently.

Text File Storage:

- Using text files for data storage simplifies the implementation and keeps the app lightweight. This approach is particularly suitable for a mobile app, ensuring it runs smoothly even on lower-end devices.

GUI Frameworks:

- GUI frameworks like Qt make it possible to create a polished and professional-looking interface. Despite using text files for storage, the app can still provide a seamless and user-friendly experience.

SCALABILITY:

- User Base Expansion: As the app gains popularity, it can be scaled to include more users by optimizing the backend and using cloud services.
- Feature Expansion: Additional features like offline mode, cloud backups, and push notifications can be added to enhance the app’s functionality over time.
- Multilingual Support: To reach a wider audience, the app can be made available in multiple languages.

IMPACT:

Educational Impact:

- Accessibility: The mobile nature of the app makes it easier for students to access and use the app regularly, leading to better study habits and improved academic performance.
- Real-Time Tracking: Students can log their study hours in real-time, ensuring more accurate tracking and feedback.

Social Impact:

- Community Building: Mobile platforms enable easy sharing and collaboration features, fostering a strong community of learners.
- Inclusivity: A mobile-first approach ensures that even those without access to desktops can benefit from the app.

Economic Impact:

- Market Reach: A mobile app can reach a broader audience, increasing the potential for monetization through premium features, ads, or in-app purchases.
- Sponsor Engagement: Companies can sponsor the app, gaining exposure to a young, tech-savvy audience, while also contributing to educational initiatives.