CCEXTRACTOR DEVELOPMENT

Ultimate Alarm Clock II - Flutter App

About Me:

Name: Deepanshu Singh
University: ITM University, Gwalior, India
Github: Deepanshuigtm (Deepanshu Singh) · GitHub
LinkedIn: https://www.linkedin.com/in/deepanshu-singh-6a57b1235/
Resume: https://drive.google.com/file/d/1IWSMyu32LJUFvA5__nAdFmPO8WQZyo-9/view
Location: Gwalior, Madhya Pradesh, India
Email: 2207deepanshu@gmail.com
Phone no: +91 8103702636
Timezone: IST(GMT +5:30)

Project: Ultimate Alarm Clock II | CCExtractor
Proposal: https://github.com/Deepanshuigtm/ccextracter-gsoc-proposal

Abstract:

This project aims to build an Alarm Clock app that develops an innovative alarm clock with intelligent functionalities. This advanced clock is designed to intuitively dismiss alarms based on user activity on their phone, **real-time weather conditions**, and additional customizable parameters. It introduces various challenges to ensure users are fully awake, offering a unique approach to morning routines. Moreover, it incorporates a feature **allowing users to set shared alarms**, promoting collaboration and coordination among multiple individuals. This project **aims to redefine the traditional alarm clock experience, enhancing wake-up routines with modern technology and personalized settings**

Goals:

Here are some goals for the project:

- Enable users to **create profiles** with different alarm settings for various scenarios, including **workdays, weekends, and vacations**
- Users can **add alarms to specific profiles**, such as assigning an alarm to the working days profile if it's suitable. Additionally, **they can remove alarms from a profile if needed**, providing flexibility in managing alarm configurations according to their preferences .

- I**f there are no violations of YouTube's Terms of Service**, then I want to create a new feature where **users can search for their favorite music** using the youtube_explode Flutter package. They can then set the searched music as their alarm ringtone
- **Dashboard feature** will provide users with a comprehensive **overview of their alarm usage statistics on a weekly or monthly basis**, allowing them to track their adherence to set alarm plans and analyze their snooze habits
- **friend system** where users can add friends. **Weekly leaderboards showcase friend scores**, fostering friendly competition and motivation for improvement. **Users can analyze weekly performances** to enhance their habits and achieve personal goals.
- **Implement Firebase Cloud Messaging for notifications**: **alarms with 10-minute reminders** and redirect to alarm upon click, weekly and monthly **report notifications**, and **weekly leaderboard showcasing top performers among friends**, elevating user engagement and performance monitoring.
- Setup **automatic CI/CD** on Github
- Setup **issue and PR template** on Github

Implementation:

App implementation is divided into the following **Milestones**

(all dated mentioned below are of 2024)

- **Community bonding period (April 19 to May 20)**
    - Interact and engage in the community, get to know my mentors and fellow students. Also, I would try to learn about other ongoing projects in the organisation, especially those related to flutter.
    - I will be discussing the **app flow** in detail with mentors. I would go with UI that is intuitive and easy to use and app flow that is efficient as much as possible.
    - Since this app would be published on the **Google Playstore**, I would start preparing assets like icons.

- **MILESTONE 1 (June 27 to July 12)- Create profile setup**

    - Users can create a new profile by completing a form, providing details such as the **profile title (e.g., "Weekend Alarm")** etc . This information will be stored in the database.
    - Then Users can **add alarms to these profiles by accessing the alarm section**, where all alarms are listed on the home page. From there, they can select an alarm and add it to one of their profiles
    - To facilitate editing of profiles, distinct methods will be employed. Adding or editing profiles will involve **creating a new collection in Firestore specifically for profiles**. Each profile will contain alarm IDs associated with it

- **Schema of User Profile**
  - The user profile schema will include a **unique automatically generated ID** from the database, ensuring each profile is uniquely identified within the database.
  - Schema will include a **title field of type string**, allowing users to input a descriptive title when creating a profile.
  - **Description field, which will be optional.** Users can choose to provide additional details about their profile if they wish.
  - The **isShareable** field will be of type boolean, allowing users to specify whether they **want the profile to be shared or not**.
  - Within each profile, there will be another **collection named "alarmsProfile,"** which will **contain all the alarm IDs that the user has added to that specific profile.**

- **MILESTONE 2 (June 27 to July 12)- Dashboard feature**
  - Design the **UI layout for the dashboard,** incorporating **graphs to visualize** day-by-day user progress with respect to alarm usage
  - Implement a graphical representation, such as line **charts or bar graphs**, to display alarm usage data over time.
  - Utilize Flutter widgets like charts or custom drawing to create interactive and visually appealing graphs**.**
  - Integrate **data retrieval and processing functionality** to fetch user alarm usage statistics from the database.
  - Populate the graphs dynamically with the retrieved data, **allowing users to track their progress** and adherence to alarm plans
  - Ensure responsiveness and usability of the **dashboard UI across different screen sizes and orientations.**
  - Test the dashboard UI extensively to verify its functionality, visual appeal, and performance**.**

- **MILESTONE 3 (July 12 to July 26)- Friend system and weekly leaderboard feature**
  - **Friend system Implementation**
    - Develop a **user interface for adding friends**, allowing users to search for and send friend requests to other users.
    - Implement **backend functionality** to manage friend requests, including sending, accepting, rejecting, and canceling requests.
    - Create a **user profile page where users can view** their friends list and pending friend requests.
    - Integrate functionality to **display friend profiles and dashboards, allowing users to view their friends' progress and statistics**

- Ensure data privacy by **implementing appropriate access controls**, allowing users to set visibility preferences for their **dashboard and restrict access to certain friends if desired.**
- **Implement real-time updates** to reflect changes in **friends' progress and statistics**, providing users with up-to-date information.
- Test the friend system thoroughly to ensure seamless user experience and address any potential issues or bugs.

  - **Leaderboards feature**
- Design a **leaderboard UI displaying all of his friends scores** at the end of each week and Implement backend functionality to calculate and update friend scores based on their weekly performance
- Create interactive elements **allowing users to view detailed weekly performance** metrics for each friend.
- Provide **feedback mechanisms** allowing users to track their progress and compare their scores with friends.
- Continuously optimize the leaderboard functionality based on user feedback and usage patterns.

- **Optional MILESTONE 4 (July 26 to August 21) - Firebase Cloud Messaging for notifications**

  - **Implement alarm notifications**: Create a service to send alarm notifications with 10-minute reminders. On click, redirect users to the alarm screen.
  - **Create weekly and monthly report notifications**: Develop services to send notifications for weekly and monthly reports summarizing user performance. Allow users to customize notification settings for these reports.
  - **Implement leaderboard notifications**: Set up a service to send notifications for the weekly leaderboard showcasing top performers among friends. Users can customize notification settings for this feature.
  - **Create notification settings**: Develop a user interface for notification settings where users can toggle on/off for each notification type (alarms, weekly reports, monthly reports, leaderboard).
  - **Store user notification preferences**: Save user notification preferences in the database to apply them when sending notifications.
  - **Handle notification preferences**: Modify notification sending logic to respect user preferences. Only send notifications for enabled types.
  - **Test notification functionality**: Thoroughly test notification sending and handling, ensuring that users receive the correct notifications based on their preferences.
  - **Optimize notification delivery**: Continuously monitor and optimize notification delivery to ensure timely and reliable notifications.

- **Optional MILESTONE 5 (August 22 to September 10) - YouTube Music Alarm Integration**
  - Integration of **youtube_explode** Flutter package: Incorporate the youtube_explode package into the project to enable seamless interaction with YouTube's API for music search functionality.

- **Implement YouTube music search**: Develop a search feature within the app, utilizing youtube_explode to **fetch titles, artists, and video IDs of searched music**.
- Design user-friendly search interface: Create an intuitive and visually appealing user interface (UI) for the music search functionality, ensuring ease of use for users.
- **Enable music selection for alarms**: Allow users to select their preferred music from the search results and set it as their alarm ringtone within the app.
- **Enhance alarm customization**: Provide users with the option to personalize their wake-up experience by waking up to their favorite tunes sourced directly from YouTube.
- **Ensure reliability and efficiency**: Test the integration thoroughly to ensure smooth functionality and optimize performance for a seamless user experience.

- **Optional MILESTONE**
  - **Automatic CI/CD for repo for releasing builds**
    If i get free time between i will also work on setting up ci/cd for the repo by **Github Actions** for automating many chore tasks like releasing builds, siging up apks, etc. There are various tools listed on flutter deployment doc like codemagic and fastlane that could be used for setting up the workflow. Appropriate tools and actions would be implemented after discussion with mentor
    -
  - **PR and issue template for bugs, features or feedback.**
    User can **report bugs, request and give feedback** by ready-made templates on Github. These templates auto-populate the issue/pull-request description field and provide a skeleton framework that contributors can fill out. They help provide a baseline standard of informational quality and organisational rigour.

My background / Technical skills

I am **Deepanshu Singh**, **3nd year computer science student at ITM University, Gwalior**, India. I've been enjoying web development for the first time ever since I built my **first Python project, "Automation"**. I have been deeply immersed in web development. Initially, I **began by exploring Flutter** and Python, with my early Flutter projects involving the creation of basic Android **applications in Android Studio**, such as a tic-tac-toe game, quiz app etc.

In my 10th grade, I built many small applications in Python, such as a Stock Portfolio Tracker and a Fitness Tracker. I used to post these **projects on social media** to show my friends and family.Then I became interested in **mobile application developmen**t and built many projects in Java. Additionally, I developed native applications using Flutter. After that, I learned Android development, web development, **Flutter, React, Docker, Django, Flask, and shell scripting**. I began participating in Capture **The Flag (CTF) contests** during one of my college events

Right now I am proficient in **app dev (both native and framework like react and flutter), and full-stack web dev (js, HTML, Django, react, flutter).** I use vscode, android studio and **sublime** as my text editors. Recently i tried learning server hosting, cloud and firebase.

I have gained experience **working with several startups as an intern**, where I contributed to various projects involving the development of mobile applications, web applications, and more. Recently, I have focused on a **Flutter project** where I delved into various state management techniques such as **Bloc and Riverpod** to streamline state management efficiently. Additionally, I have acquired expertise in cloud technologies like **Firebase and Firestore** during my recent projects.

I worked as **Python Automation** Intern in CHAKRABORTY LAKHAR INNOVATIONS PVT.LTD, used technologies like Web scraping using Python, **BeautifulSoup and Selenium** for automation . There I learned agile methodologies to design, **software testing** and debug software applications
Certificate of Internship - Link of certificate
Letter of recommendation - Link of letter of recommendation

I am **Microsoft Learn Student Ambassador**, helping students of my college. The MSP program enhances students' employability by offering training in skills not usually taught in academia, including knowledge of Microsoft technologies.
I am also a coordinator, responsible for managing, ideating and supervising most things tech in the college. I worked on various campus development initiatives.
My recent focus is on the **TecOS** initiative, **encouraging the open-source culture in our college**.
I am very active in open source. I try to participate in many of the open-source events happening. I participated in **KWOC, DWOC, and Hacktoberfest and this is my attempt in GSOC.** I am also actively involved in encouraging open-source culture in my college, I listed a few initiatives like org review, human of open source, etc under TeCOS.

AI Task Automation 2024
I have developed an **AI-based web application** that assists users in a wide range of tasks. Users can ask questions by providing context through various media formats such as video files, audio files, PDFs, or text documents. Additionally, the application offers numerous other features aimed at saving users' time and enhancing their productivity.
Containerize Next.js, Flask with Docker. Use Docker Compose for multi-service setup. Optimize images, manage networking. Simplify deployment and scaling
https://github.com/Deepanshuigtm/ml-app.git

Personal Projects -
https://github.com/Deepanshuigtm/internshala_job.git
https://github.com/Deepanshuigtm/flutter-chat.git
https://github.com/Deepanshuigtm/webscraping_amazon_bags.git
https://github.com/Deepanshuigtm/guess_My_Number.git

Contributions to open source:

CCExtractor Ultimate Alarm Clock

Some of my contribution to **Ultimate Alarm Clock** project are listed below :

- **Fixed a bug that was reported by someone else**. The issue can be found here: https://github.com/CCExtractor/ultimate_alarm_clock/issues/170
- **Feature Swipe-to-Delete alarm** - https://github.com/CCExtractor/ultimate_alarm_clock/issues/271
- **Undo option Swipe-to-Delete alarm -** https://github.com/CCExtractor/ultimate_alarm_clock/issues/314
- **Customizable Undo Duration -** https://github.com/CCExtractor/ultimate_alarm_clock/issues/378

Commitments:

After June, I'll begin my last year of college, the fourth year, and will dedicate full-time hours to the project. Occasionally, I'll need to attend college, but apart from those scheduled days, I'll be fully committed. In the event of any emergencies, I'll inform my mentor in advance.

Weekly Commitment: 40-50 hours

I am more than ready to work well past my committed time if needed.