

Fit Life Tracker App

Name: Aniket Bourasee

Email: aniketbourasee007@gmail.com

Project Overview :

The **Fit Life Tracker App** is an Android application designed to help users track their daily fitness activities like walking, running, swimming and cycling. Users can create and manage their fitness profiles, monitor progress, and receive notifications to stay motivated. This app provides insightful data about daily, weekly, and monthly activity logs and allows users to set and track fitness goals.

Features :

1. Activity Tracking

- Track daily steps, distance covered, and calories burned.
- Allow users to select the type of activity: walking, running, or cycling.
- The app calculates fitness metrics like distance and calories based on user activity data.

2. User Profile :

- Users can create a profile with personal information such as:
 - Name
 - Age
 - Weight
 - Height
- The app automatically calculates and displays the user's **Body Mass Index (BMI)** based on their profile.

3. Progress Monitoring :

- The app tracks and displays user progress in daily, weekly, and monthly charts.
- Visual summaries (e.g., bar charts) provide insights into steps taken, calories burned, and distances covered.
- Users can view achievements or milestones when certain fitness goals are met (e.g., walking 1000 steps in a day).

4. Data Persistence :

- The app uses **SQLite** or **Room database** to persistently store:
 - User profiles
 - Daily activity logs (steps, calories, distance, etc.)
- Data is retained even when the app is closed and reopened.

5. Notifications :

- The app sends daily reminders to users to track their activities.
 - Users can customize reminder timings based on their schedule.
-

Technical Aspects :

1. Android Framework :

The app is built using the Android Framework with Java. It follows best practices for mobile app development and adheres to Material Design guidelines for an intuitive and user-friendly interface.

2. Libraries and Tools :

- **Room or SQLite:** Used for data persistence (user information, daily activity logs).
- **MPAndroidChart:** A library used for generating charts and graphs to display user progress.
- **RecyclerView:** Displays historical fitness data in a list format.

3. Android Permissions :

The app requires specific Android permissions to function correctly:

- **Notification Permissions:** For sending reminders to the user to track their fitness activities.
- **Storage Permissions:** To store data persistently using SQLite/Room database.

4. Input Validation

The app incorporates input validation to ensure data accuracy, such as:

- Proper validation for user profile details (name, email, age, weight, etc.).
 - Validation to avoid setting activities in the past or invalid times.
-

Usage :

1. User Profile Setup :

- On first launch, the user is prompted to create a profile by entering their name, age, weight, and height.
- The BMI is calculated automatically and displayed in the profile section.

2. Tracking Activities :

- Select an activity (walking, running, swimming and cycling) and start tracking your fitness journey.
- For outdoor activities, the app uses Google Maps to track the route in real-time.

3. Monitoring Progress :

- View daily, weekly, and monthly progress in the form of charts and statistics in the **Progress** section.
- Achievements will appear as milestones when specific goals are met.

4. Reminders :

- Daily reminders are sent to encourage the user to continue tracking their fitness activities.
-

Project Structure

AndroidManifest.xml:

- Contains permissions (e.g., notifications, vibration).
- Declares all app activities (SplashActivity, MainActivity, etc.) and the AlarmReceiver.
- Specifies app settings like backup rules, themes, and launcher activity.

build.gradle (Module: app):

- **Plugins:** Includes Android application plugin.
- **Android Configuration:**
 - Specifies the app's namespace, compile and target SDK versions (34), minimum SDK (24), and app version (1.0).
 - Configures build types (release with no code shrinking).
 - Sets Java compatibility for version 1.8.
- **Dependencies:**
 - Includes libraries like AppCompat, Material Design, Room (for SQLite DB), MPAndroidChart (for charts), and testing frameworks (JUnit, Espresso).
 - Local dependencies for MPAndroidChart are added through .jar files.

ActivityDao.java:

This interface is a Data Access Object (DAO) for managing Activity entity operations in the Room database. It includes methods for:

- **insertActivity(Activity activity)**: Inserts a new activity record.
- **getAllActivities()**: Retrieves all activities, ordered by timestamp (descending).
- **getActivitiesBetween(long start, long end)**: Fetches activities within a specified time range.
- **deleteAllActivities()**: Deletes all activity records from the database.

AppDatabase.java:

This class defines the Room database for the Fitness Tracker app, managing data storage and access. Key features include:

- **Entities**: Includes User and Activity entities representing user profiles and fitness activities.
- **DAOs**: Exposes UserDao and ActivityDao for interacting with their respective database tables.
- **Migration**: Handles schema updates from version 1 to version 2, specifically by adding a distanceCovered column to the Activity table.
- **Singleton Pattern**: Implements a thread-safe singleton pattern to ensure a single instance of the database.

Activity.java:

This class defines the Activity entity, representing a user's fitness activity record in the database.

- **@Entity:** Marks the class as a Room entity with the table name "activity".
- **Fields:**
 - **id:** Auto-generated primary key to uniquely identify each activity.
 - **activityType:** Type of activity (e.g., walking, running, cycling).
 - **stepsCount:** Number of steps recorded during the activity.
 - **caloriesBurned:** Calories burned during the activity.
 - **distanceCovered:** Distance covered in meters (added in version 2).
 - **timestamp:** Time when the activity was logged

User.java:

This class defines the User entity for the database.

- **@Entity:** Marks it as a Room entity with the table name "user".
- **Fields:**
 - **id:** Auto-generated primary key.
 - **name:** User's name.
 - **age:** User's age.
 - **weight:** User's weight in kg.
 - **feet:** User's height in feet.
 - **inches:** User's height in inches.
- **Constructor:** Initializes a new User with specified details.

UserDao.java:

This interface defines the DAO for the User entity, enabling user data interaction in the Room database.

- **@Dao**: Indicates it serves as a DAO for User operations.
- **Methods**:
 - **insertUser(User user)**:
 - **@Insert**: Adds a new User to the database.
 - **getUserById(int userId)**:
 - **@Query**: Retrieves a User by its unique ID.
 - **getAllUsers()**:
 - **@Query**: Returns a list of all users in the database.

ActivityAdapter.java:

This class is the adapter for displaying activity data in a RecyclerView.

- **Fields**:
 - List<Activity> activities: Holds activity data.
- **Constructor**:
 - ActivityAdapter(List<Activity> activities): Initializes the adapter with activity data.
- **Methods**:
 - onCreateViewHolder(): Inflates the layout from item_activity.xml and returns a new ActivityViewHolder.
 - onBindViewHolder(): Binds activity data to the ViewHolder at the specified position.
 - getItemCount(): Returns the total number of activities.
- **Inner Class**: ActivityViewHolder:
 - Holds references to TextView elements for activity details.
 - **bind(Activity activity)**: Updates TextViews with activity type, steps count, and formatted timestamp.
 - **formatDate(long timestamp)**: Converts timestamp into a human-readable date string.

ActivityTracker.java:

Manages activity tracking in the Fitness Tracker App.

- **Fields:**

- Spinner spinnerActivityType: Selects activity type.
- Button btnStartTracking, btnStopTracking: Start/stop tracking.
- TextView tvTrackingStatus: Displays tracking status.
- ImageView activityImageView: Shows activity-related image.
- boolean isTracking: Indicates if tracking is active.
- long startTime: Records start time.
- AppDatabase db: Database for storing activity data.

- **Methods:**

- onCreate(): Initializes UI and sets button listeners.
- startTrackingActivity(): Begins tracking and updates UI.
- stopTrackingActivity(): Stops tracking, calculates metrics, and saves to the database.
- calculateSteps(): Estimates steps based on activity type and duration.
- calculateDistance(): Computes distance from activity type and steps.
- calculateCaloriesBurned(): Estimates calories burned during the activity.
- logActivity(): Creates a log entry for the activity with details.

AlarmReceiver.java:

Handles alarm and reminder notifications in the Fitness Tracker App.

- **Fields:**

- CHANNEL_ID: Notification channel identifier.
- TAG: Logging tag.
- ACTION_STOP_ALARM: Action string to stop the alarm.
- Ringtones ringtone: Manages alarm sounds.
- Handler reminderHandler: Manages reminder sound delays.

- **Methods:**
 - `onReceive(Context context, Intent intent)`: Triggered on alarm receipt; stops alarm or creates notification.
 - `createNotificationChannel(Context context)`: Sets up a notification channel; builds notification with actions and sounds.
 - `playAlarmSound(Context context)`: Plays the default alarm sound.
 - `playReminderSound(Context context)`: Plays reminder sound for 5 seconds.
 - `stopAlarmSound(Context context)`: Stops alarm sound and resets UI.
 - `stopReminderSound(Context context)`: Stops reminder sound and resets reminder UI.

ClockActivity.java:

Manages the UI for reminders, alarms, and a stopwatch in the Fitness Tracker App.

- **UI Elements:** Displays times/dates for reminders and alarms, and allows user input for descriptions. Contains buttons for actions and stopwatch controls.
- **Stopwatch Functionality:** Start, stop, and reset capabilities, updating every 10 milliseconds.
- **Permissions:** Requests notification posting permissions for API level 33 and above.
- **Date and Time Pickers:** Allows selection of times and dates, updating corresponding TextViews.
- **Set/Delete Reminders/Alarms:**
 - `setReminder()`: Validates and schedules a reminder.
 - `deleteReminder()`: Cancels an active reminder.
 - `setAlarm()`: Similar to `setReminder()`, for alarms.
 - `deleteAlarm()`: Cancels an active alarm.

- **Scheduling Notifications:** Uses AlarmManager to trigger AlarmReceiver at specified times.
- **Stopwatch Logic:** Utilizes a Handler and Runnable for real-time updates.
- **Broadcast Receivers:** Resets UI for alarms and reminders upon notification handling.
- **Utility Methods:**
 - parseDateTime(): Converts selected date/time to a Calendar object.
 - resetAlarmUI() and resetReminderUI(): Resets UI components to defaults.

MainActivity.java:

The main entry point of the Fitness Tracker App, featuring a user interface with navigation buttons.

- **UI Elements:** Four TextView buttons for navigating to:
 - **ProfileActivity:** User profile management.
 - **ProgressActivity:** Displaying fitness progress.
 - **ActivityTracker:** Tracking various fitness activities.
 - **ClockActivity:** Setting reminders and alarms.
- **Lifecycle Methods:**
 - onCreate(): Initializes the activity, sets the content view, and binds button elements.
- **Button Click Listeners:** Each button is equipped with a setOnClickListener to start the corresponding activity using Intents.

ProfileActivity.java:

Manages the user profile in the Fitness Tracker App, allowing personal detail input and BMI calculation.

- **UI Elements:**
 - EditText fields for name, age, weight, height.
 - TextView for displaying BMI result.
 - Buttons for calculating BMI and displaying user details.
 - ProgressBar for visual feedback during BMI calculation.
- **Lifecycle Methods:**
 - onCreate(): Initializes UI components and button listeners.
- **BMI Calculation:**
 - Calculates BMI using the formula $BMI = \frac{\text{weight (kg)}}{\text{height (m)}^2}$ with feedback on weight status.
- **User Details Management:**
 - Displays a dialog for viewing, updating, or deleting user details, with confirmation for deletions.
- **Feedback:**
 - Color-coded BMI results and Toast messages for user actions.

ProgressActivity.java:

Displays user activity progress in My Fitness App with a Pie Chart and RecyclerView.

- **UI Elements:**
 - PieChart for visualizing activity distribution.
 - RecyclerView for displaying activity history.
 - Spinner for filtering options.
 - Button to delete all activity records.
- **Database:**
 - Uses Room Database for activity data persistence.

- **Filtering Logic:**
 - Allows filtering by today's activities, last week, last month, and specific activity types.
- **Data Aggregation:**
 - Populates the Pie Chart with total steps for each activity type.
- **Milestone Tracking:**
 - Notifies users when significant activity milestones are reached.
- **Methods:**
 - `onCreate()`: Initializes UI and database connection.
 - `deleteAllActivities()`: Deletes all activity records.
 - `filterAndDisplayData(String filterOption)`: Filters and updates UI.
 - `setupPieChart(List<Activity> activities)`: Configures Pie Chart.
 - `checkMilestones(List<Activity> activities)`: Checks and notifies about milestones.

SplashActivity.java:

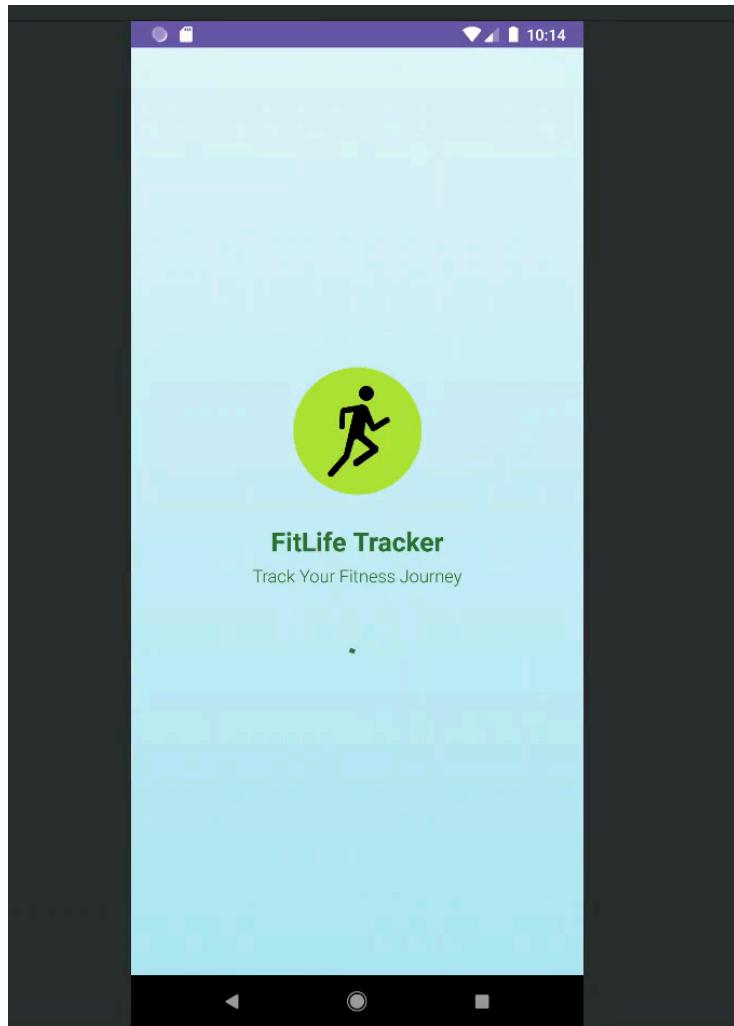
Acts as an introductory screen for My Fitness App, displaying the logo before transitioning to the main activity.

- **UI Elements:**
 - `ImageView` for the app logo.
 - **Bitmap Scaling:**
 - Scales the logo to fit the device's screen.
 - **Delay Logic:**
 - Uses a Handler to show the splash screen for 6 seconds before navigating to `MainActivity`.
 - **Methods:**
 - `onCreate(Bundle savedInstanceState)`: Initializes the activity, scales the logo, and starts a 6-second delay for the transition.
-

Fit Life Tracker App Overview

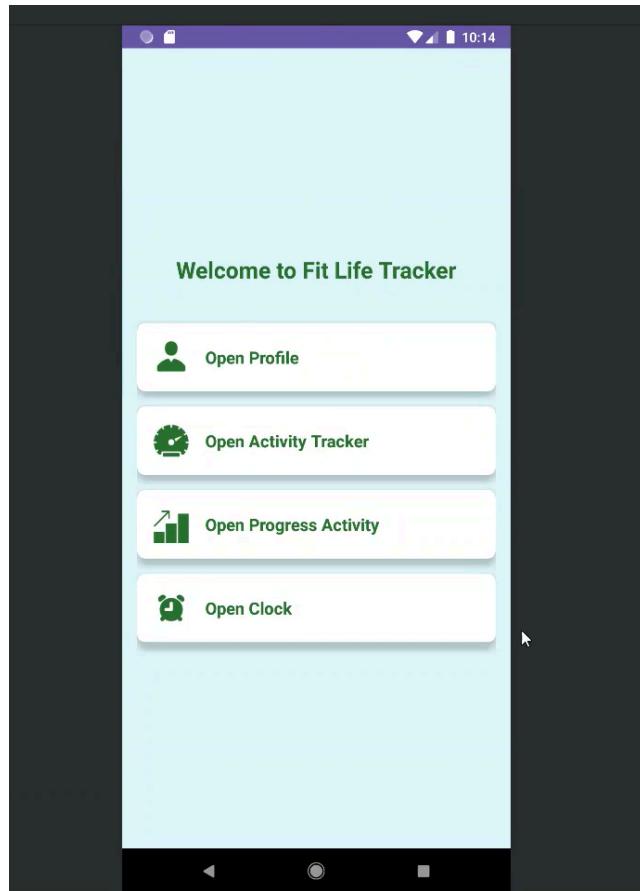
1. Splash Screen:

- The app opens with a splash screen for 6 seconds, displaying the app logo, name ("Fit Life Tracker"), and tagline ("Track Your Fitness Journey").



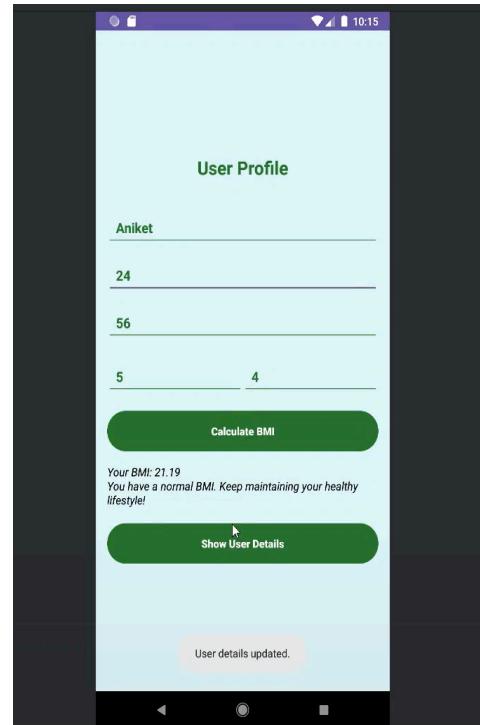
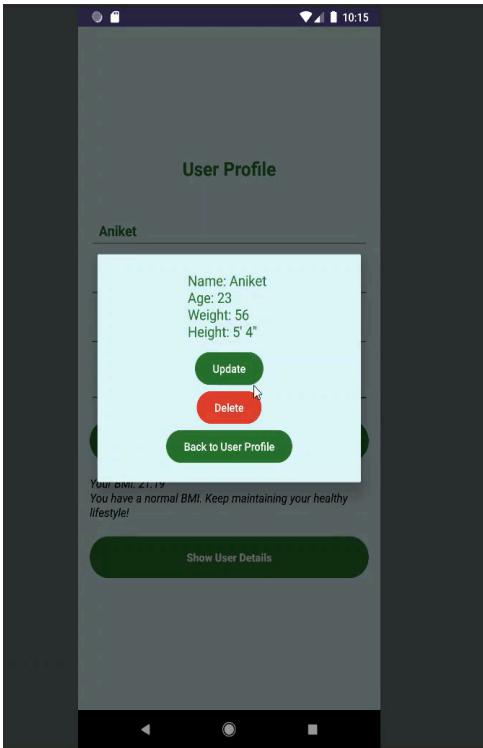
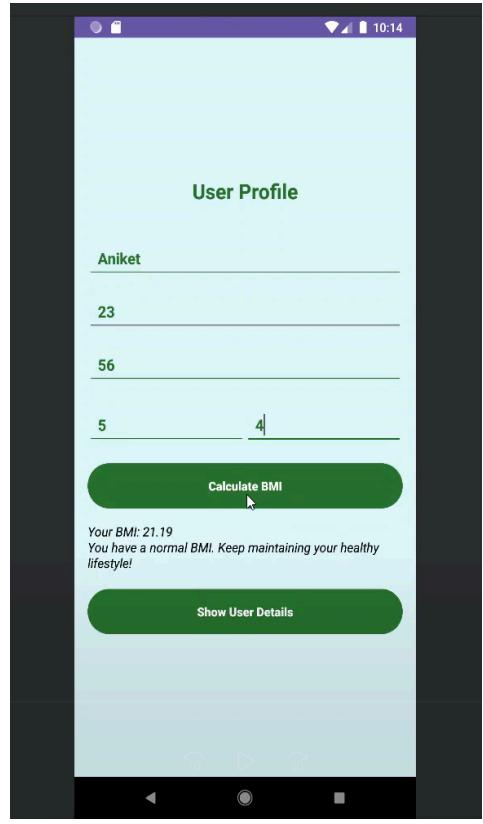
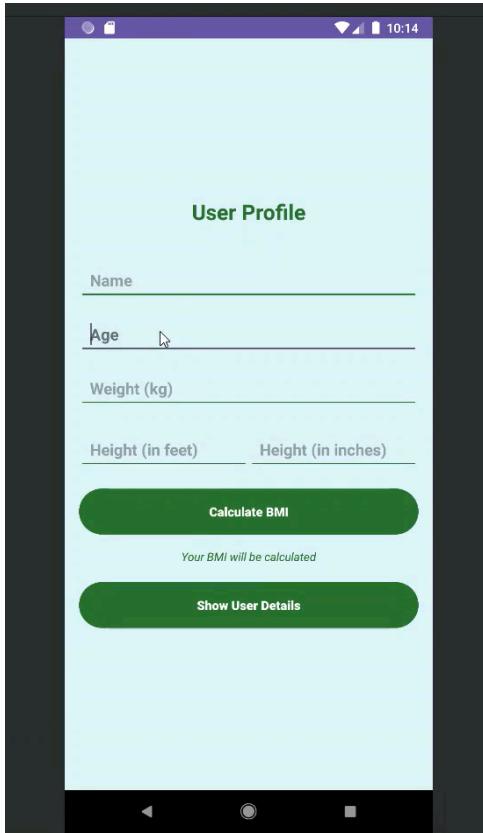
2. Main Activity:

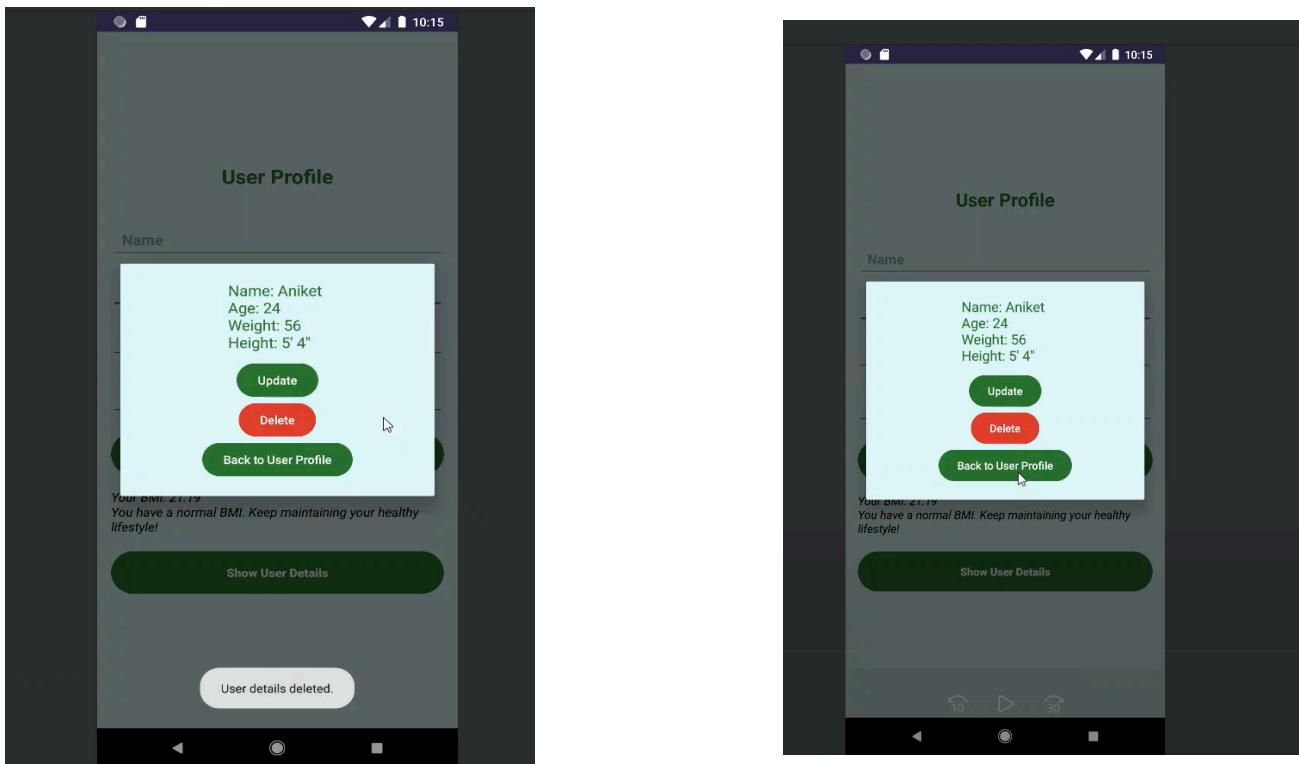
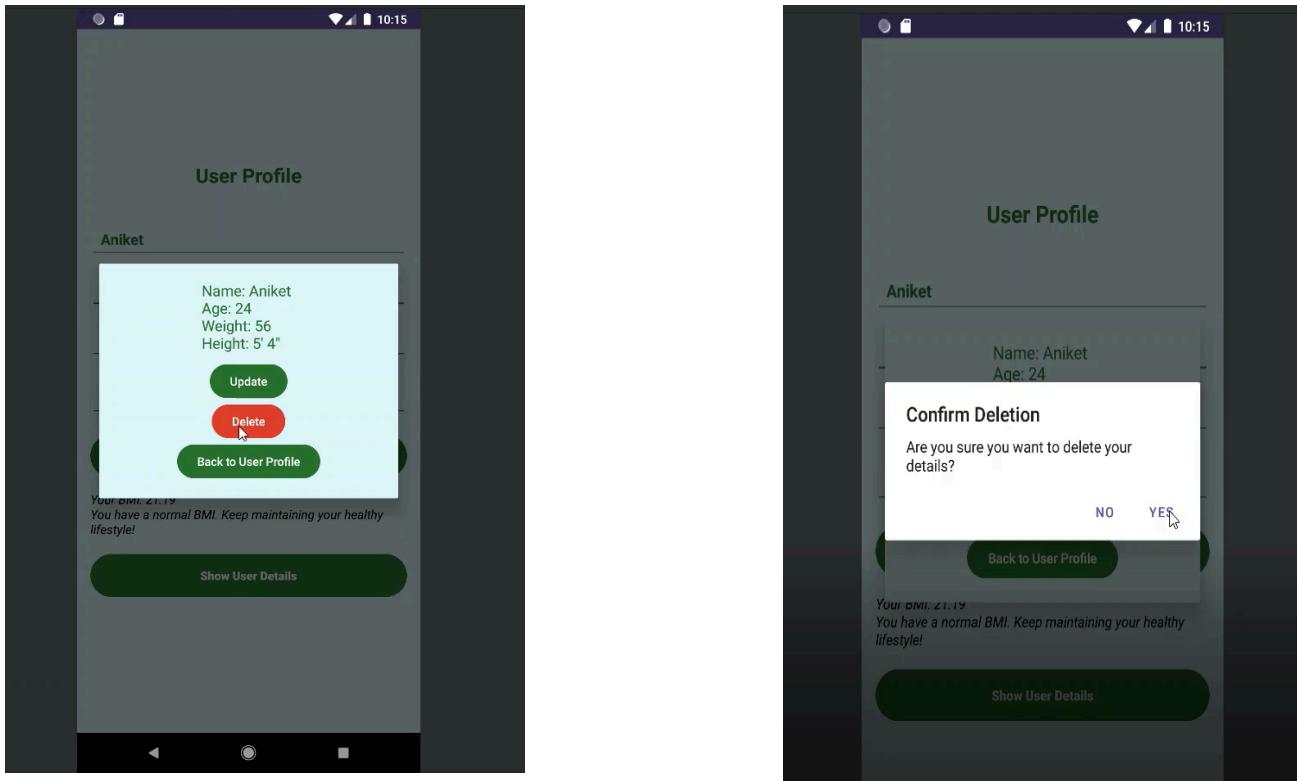
- After the splash screen, the Main Activity appears with the heading "Welcome to Fit Life Tracker."
- Four menu options are available:
 - **Open Profile**
 - **Open Activity Tracker**
 - **Open Progress Activity**
 - **Open Clock**



3. User Profile Activity:

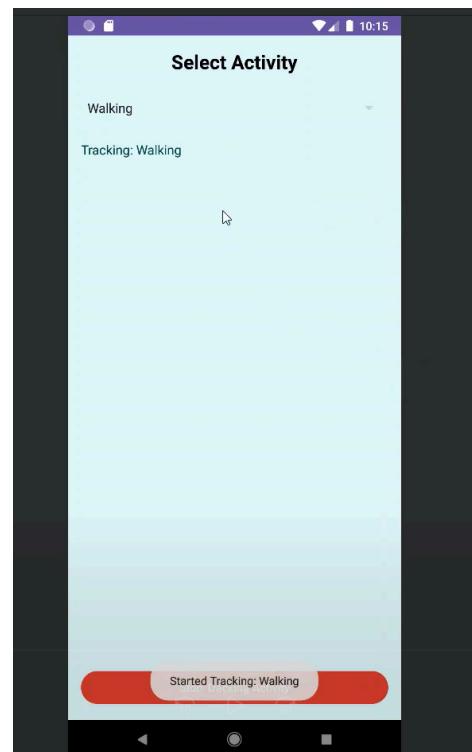
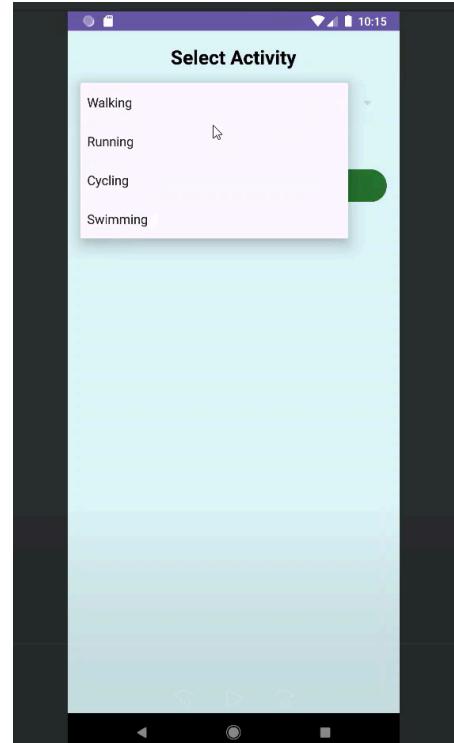
- When the user clicks on **Open Profile**, they navigate to the User Profile Activity.
- The user can input the following fields:
 - Name
 - Age
 - Weight
 - Height
- A button is provided to **Calculate BMI**:
 - When clicked, the app calculates the BMI and displays it (e.g., “Your BMI: 21.19”) with suggestions based on the result (e.g., “You have a normal BMI. Keep maintaining your healthy lifestyle!”).
- A **Show User Details** button opens a pop-up screen displaying the user's information.
- In the pop-up:
 - Buttons for **Update**, **Delete**, and **Back to User Profile** are available.
 - Clicking **Update** opens the User Profile screen again, allowing the user to modify their details.
 - After updating, clicking the update button refreshes the pop-up with updated details and displays a toast message: “User details updated.”
 - Clicking **Delete** prompts a confirmation dialog. Upon confirming, the user data is deleted, and a toast message displays: “User details deleted.”
 - Clicking **Back to User Profile** returns the user to the main User Profile Activity layout.

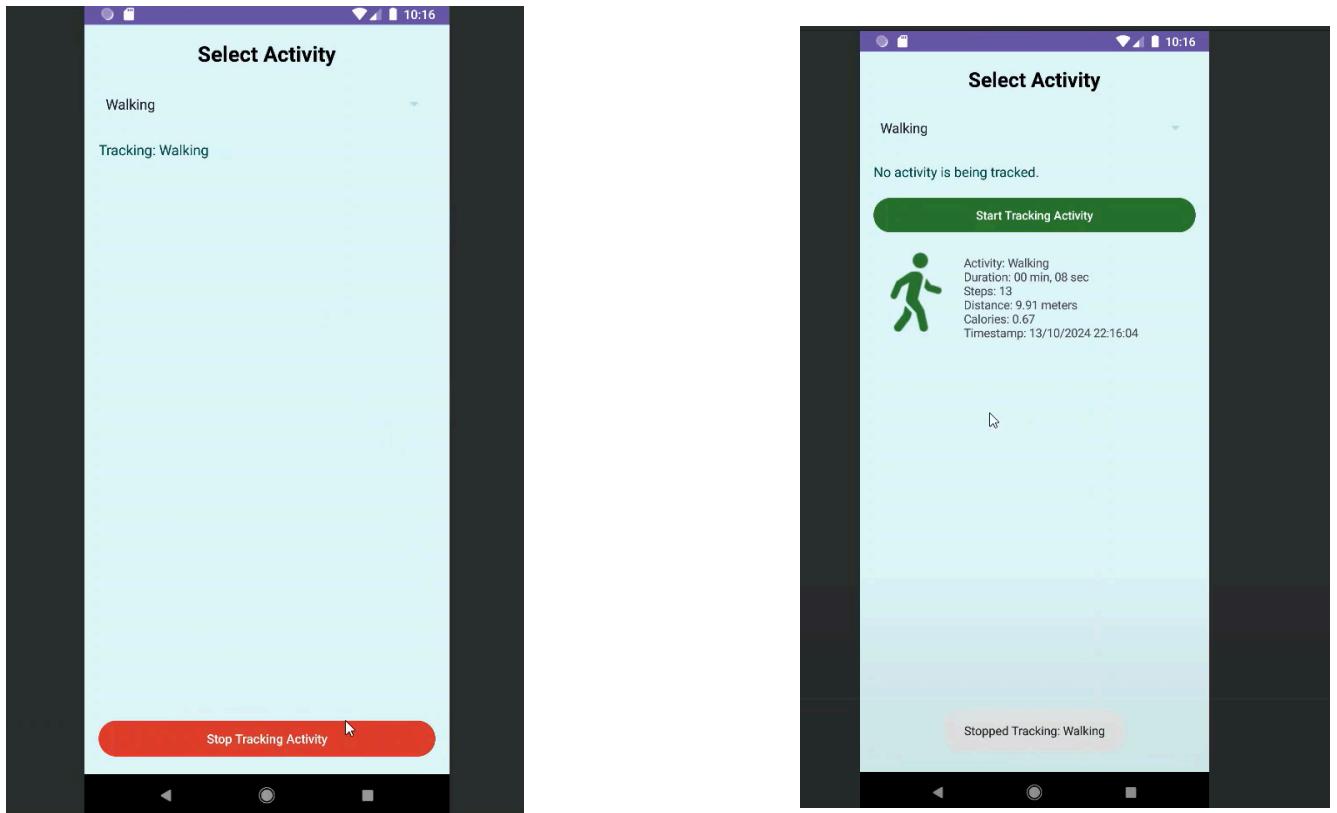




4. Activity Tracker:

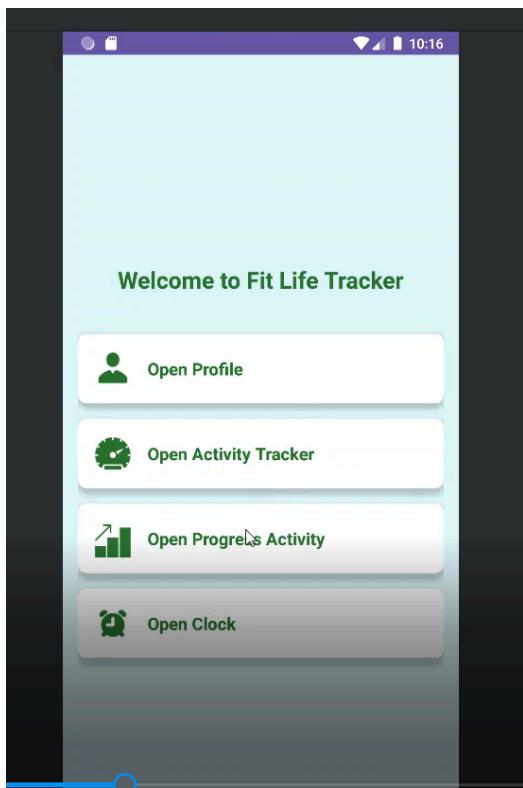
- From the Main Activity, selecting **Open Activity Tracker** leads to the Select Activity layout.
- The user can choose from four activity options:
 - Walking
 - Running
 - Swimming
 - Cycling
- After selecting an activity, the user clicks the **Start Tracking Activity** button:
 - A toast message appears (e.g., “Started Tracking: Walking”).
- A button to **Stop Tracking Activity** is available:
 - When clicked, the activity stops, and a toast message (e.g., “Stopped Tracking: Walking”) is displayed.
 - The results appear in cards showing:
 - Activity name
 - Duration
 - Steps
 - Distance
 - Calories
 - Timestamp

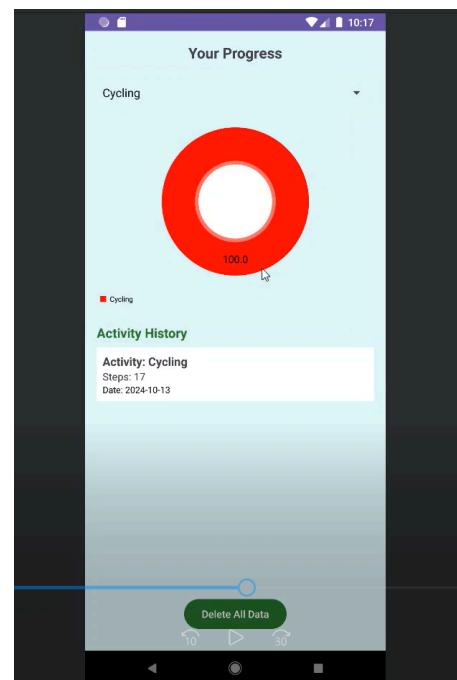
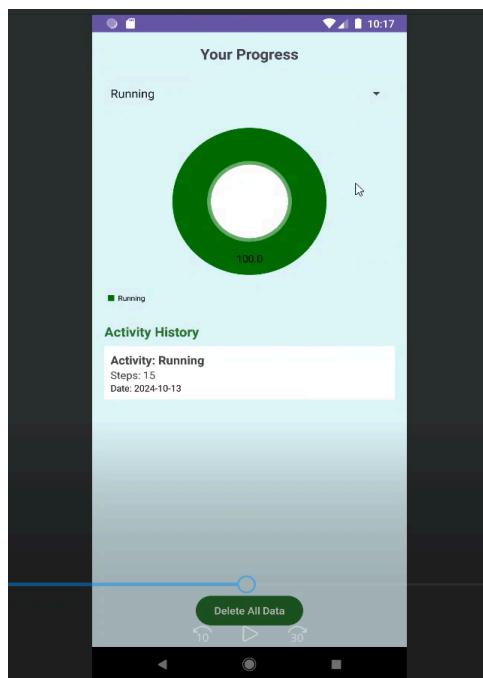
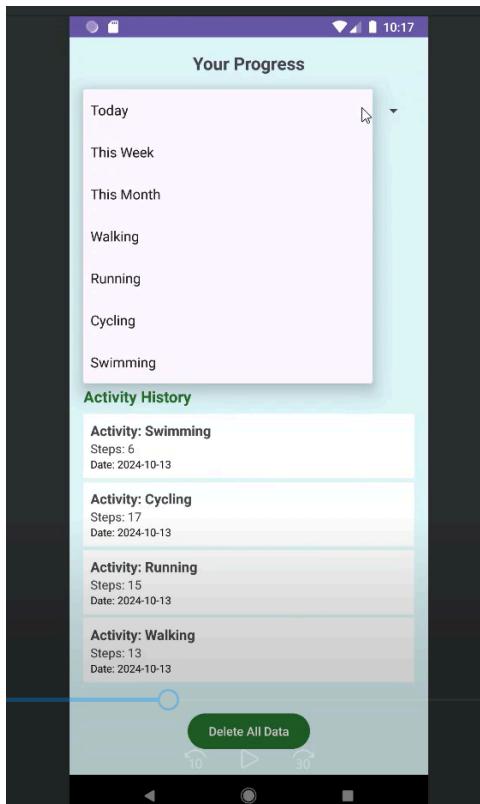


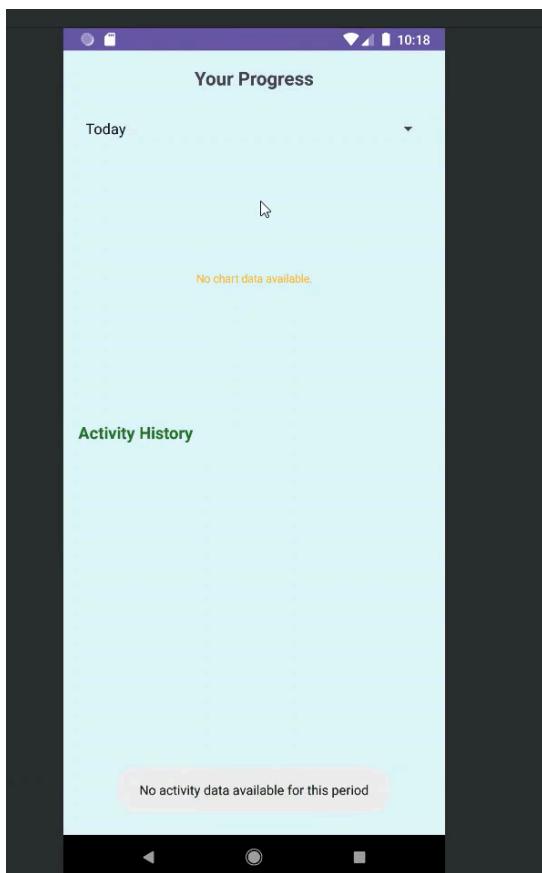
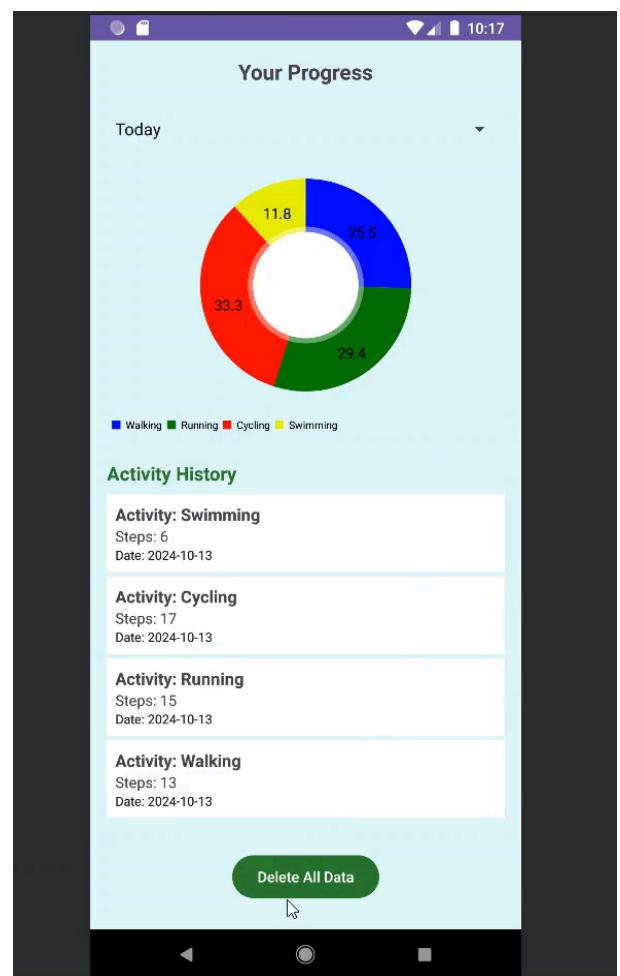


5. Progress Activity:

- Returning to the Main Activity, the user can select **Open Progress Activity**.
- This layout displays user activity progress through a pie chart.
- Users can filter their progress by:
 - Today's activities
 - This week's activities
 - This month's activities
 - Specific activity types (walking, running, cycling, swimming)
- The activity history, including details like activity name, number of steps, and date for each activity, is also available.
- A button to **Delete All Data** is located at the bottom:
 - Upon clicking, all activity data is deleted, and a toast message displays: "No activity data available for this period."

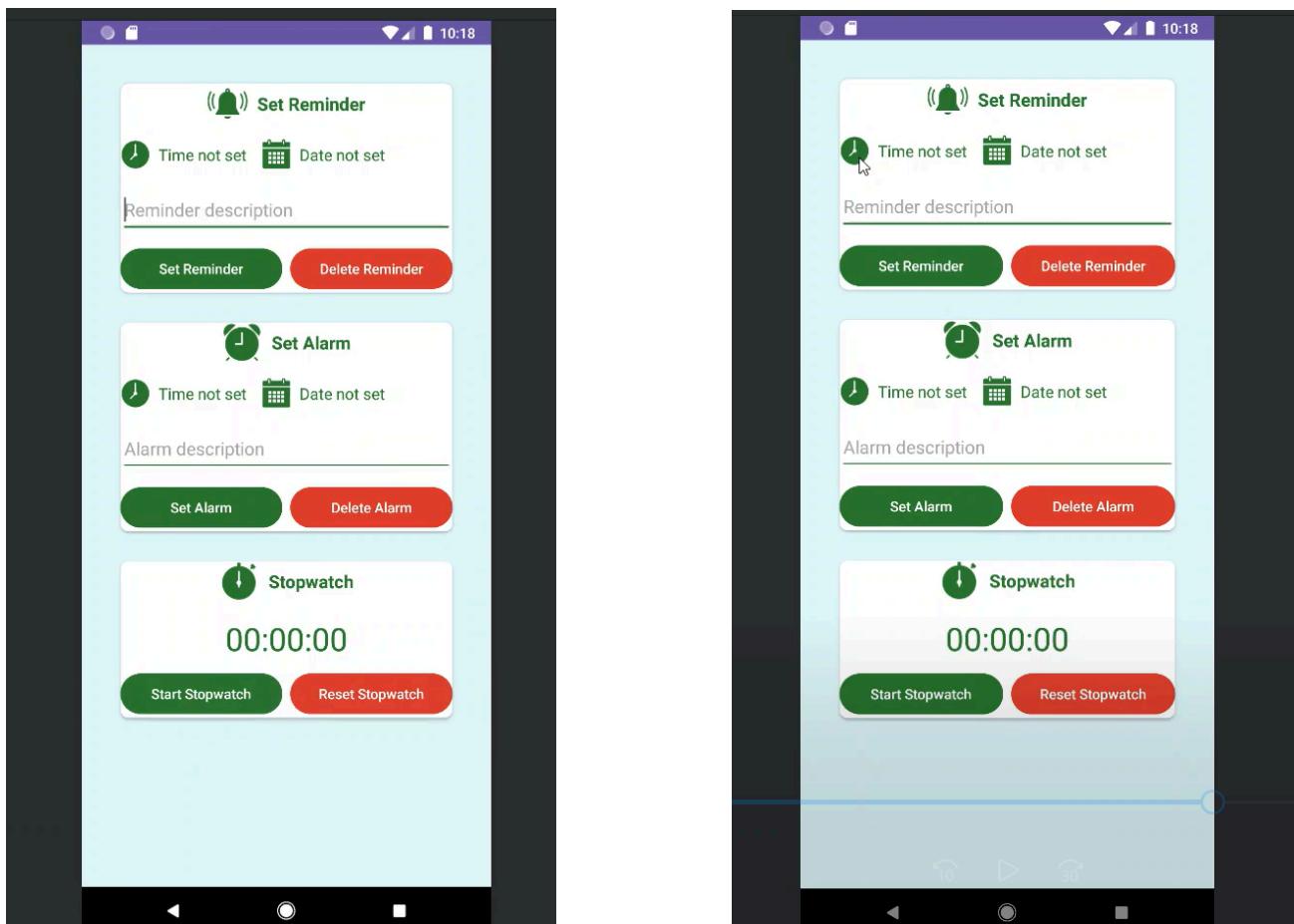


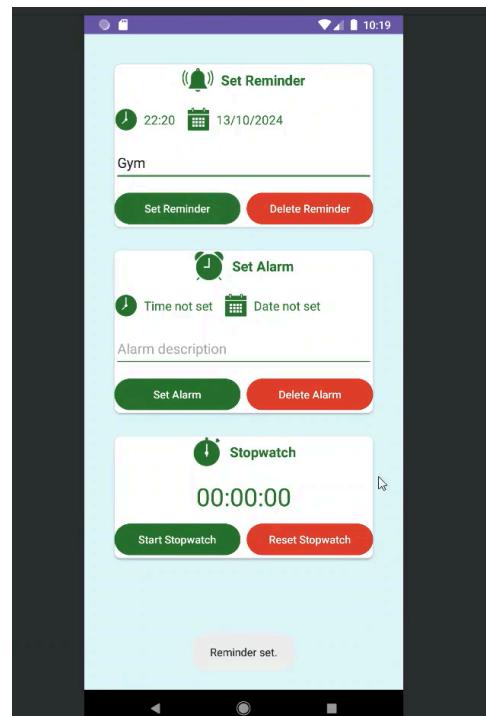
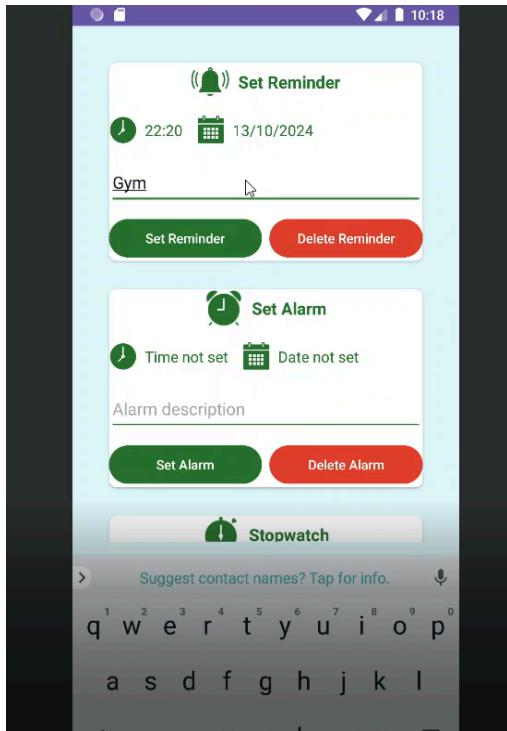
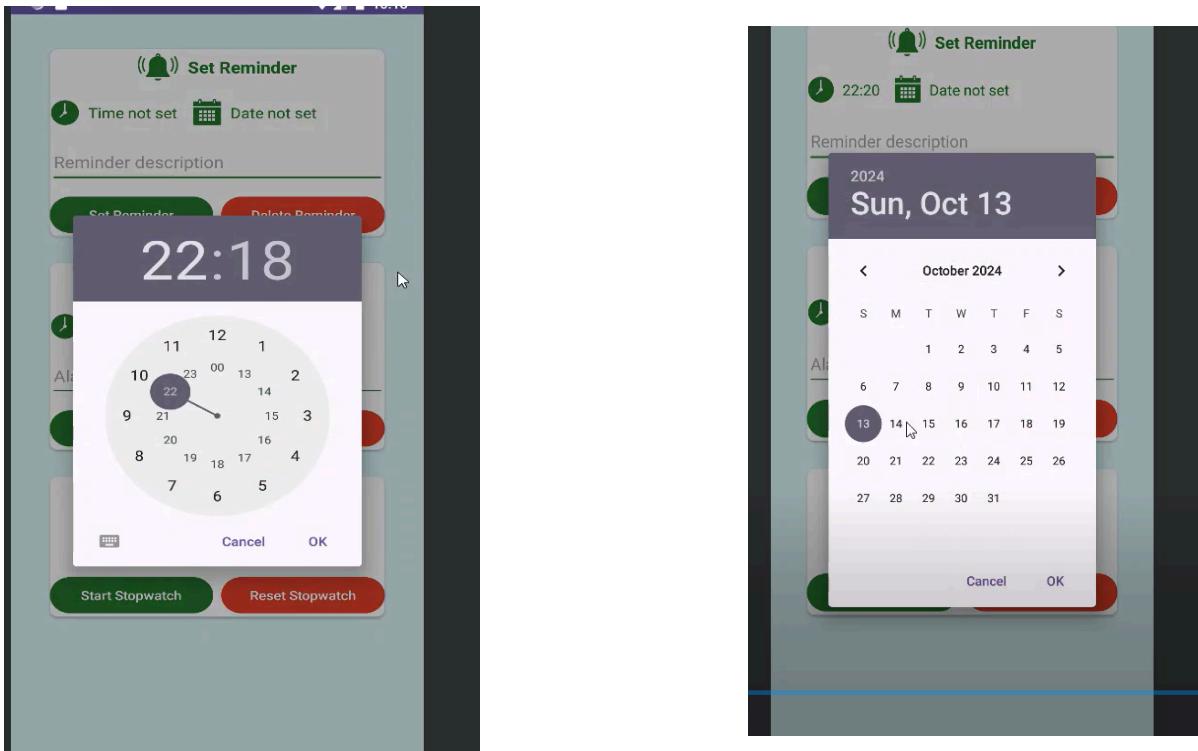


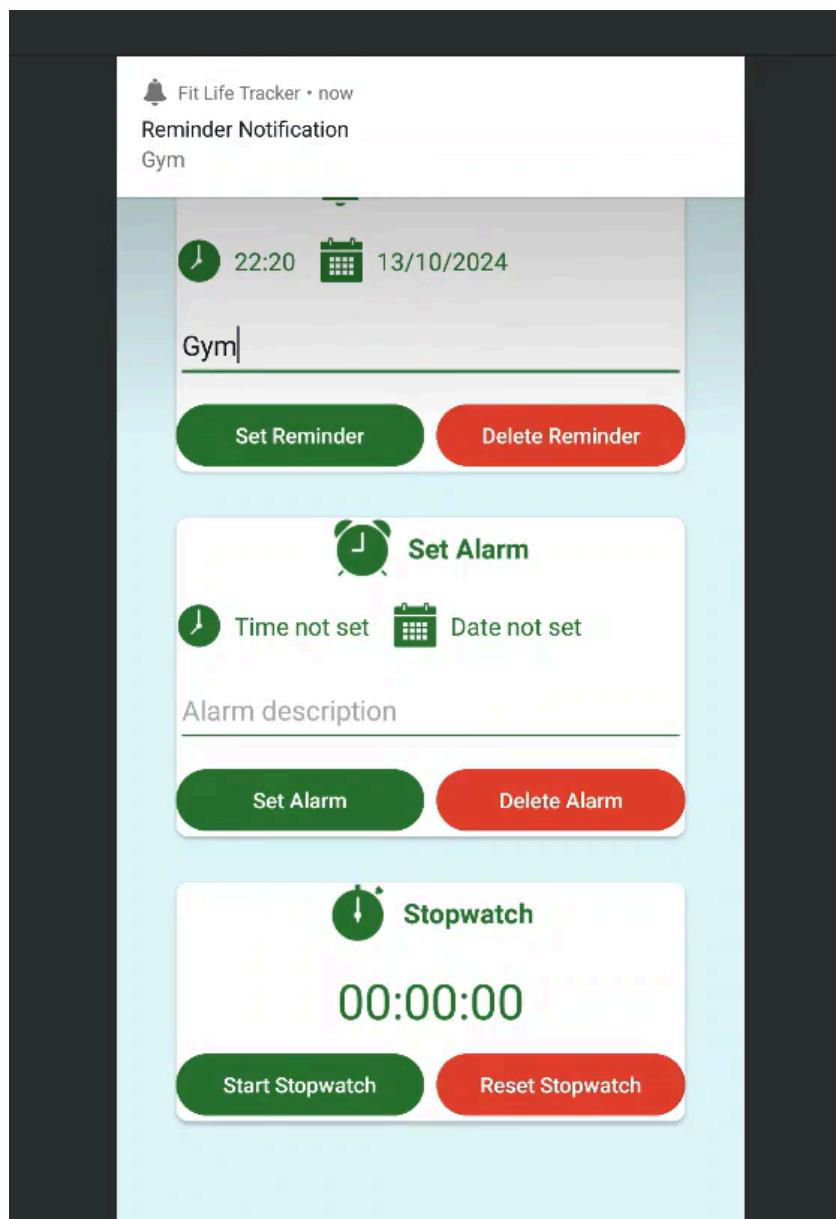


6. Clock Activity:

- From the Main Activity, selecting **Open Clock** navigates to the Clock Activity layout.
- Users can:
 - **Set Reminders:**
 - Enter time, date, and description; notifications are received after setting.

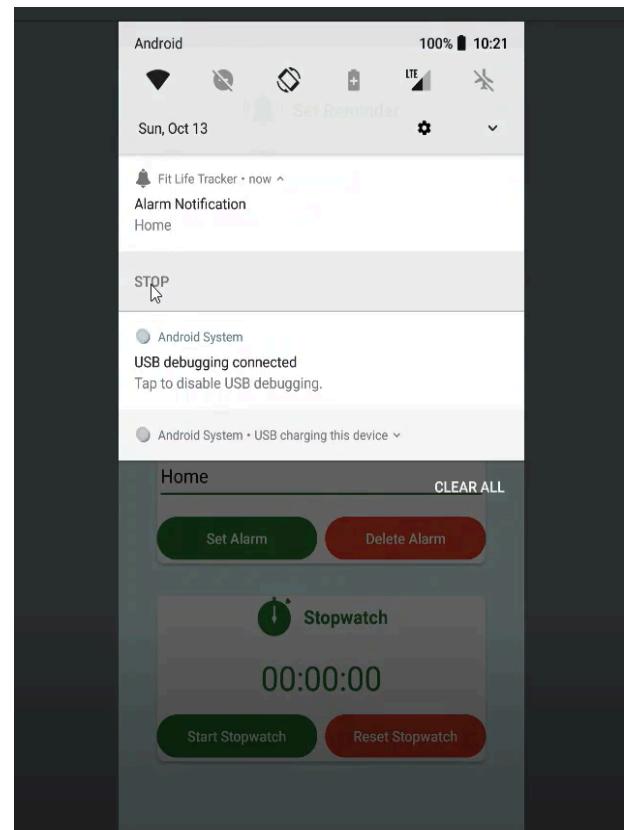
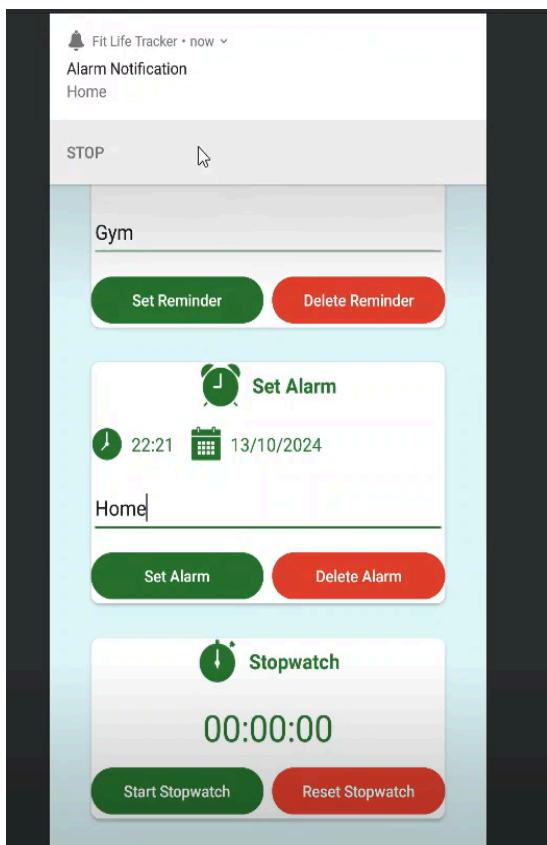
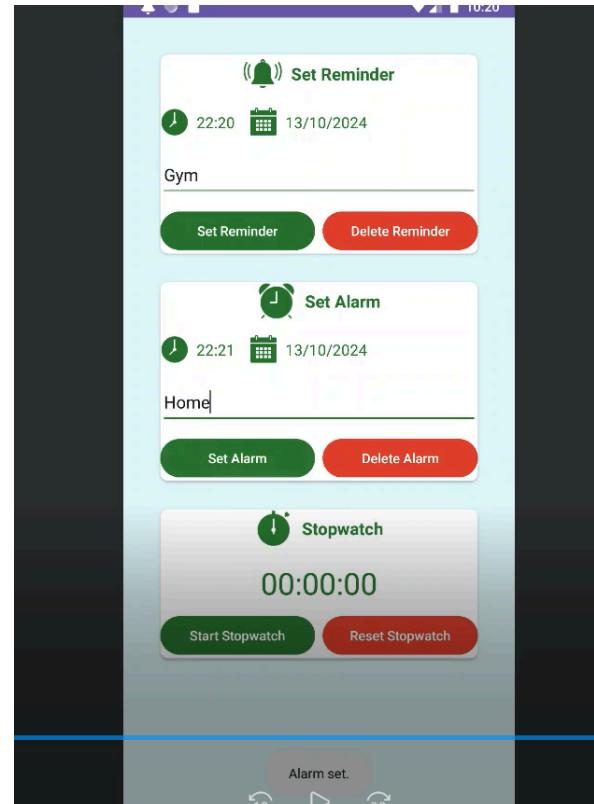
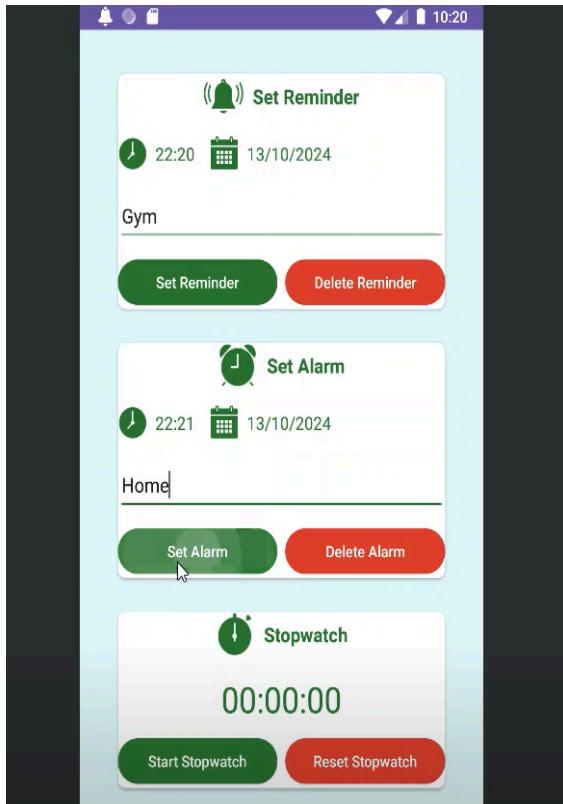






■ Set Alarms:

- Enter time, date, and description; alarms are triggered as scheduled.



■ Use Stopwatch:

- Click the **Start Stopwatch** button to begin timing.
- Stop the stopwatch using the Stop button, and reset it with the Reset button.

