**Team Project: Productivity App – Time & Habit Manager**

**Course**: Application Development 2 (Mobile)

Team Members: Reza Akbari, Muhammed Abdul-Rahmani, Stephan Shyayet Gani-Ikilama

Technologies Used: Flutter, Firebase Authentication, Firestore, Dart

**Project Overview**

The goal of this project was to develop a mobile application designed to help users manage their time and improve daily habits. The app allows users to register, personalize their preferences, and access a variety of tools such as profile management, contact options, and privacy settings. Some sections of the project were divided among team members to ensure efficient collaboration and timely completion.

**Project Development Process**

**1. Firebase Integration**

We began by setting up the Firebase backend. This included:

* Creating a Firebase project and registering our Android app.
* Adding google-services.json to the correct directory.
* Enabling **Email/Password Authentication**.
* Connecting **Firestore Database** for storing user data.

Firebase was fully connected to our Flutter project to enable both authentication and real-time data storage.

**2. UI Design and Welcome Flow**

We designed a smooth onboarding experience:

* A **Welcome Page** with an image, motivational message, and “Next” button.
* An **Interests Page** allowing users to select from predefined topics (e.g., Fitness, Study, Health).
* A **Signup Page** where users enter their name, email, and password. Age restriction was also introduced, where users under 16 are not permitted to register.
* A **Login Page** was also implemented with form validation and email verification support.

**3. Email Verification and Validation**

After successful sign-up:

* We implemented **Firebase's built-in email verification** to ensure the authenticity of user accounts.
* Users receive a verification link and cannot proceed until their email is verified.
* We used user.sendEmailVerification() and user.reload() methods to ensure the process was reliable.

**4. Home Page with Profile Display**

Once logged in:

* Users are redirected to a **Home Page** that displays their name, email, and profile image.
* A default image and edit icon were included. Users can select a profile picture from the gallery using the image\_picker package.
* We used FirebaseAuth.instance.currentUser.displayName to fetch and display the user’s name.

**5. About Page – Age & Name Handling**

The **About Page** allows users to:

* Edit their name and age.
* View their email address (read-only).
* If a user enters an age below 16, the app prevents further access and displays an appropriate warning.
* Data is saved to Firestore and the display name is updated using user.updateDisplayName().

**6. Privacy Page**

We created a **Privacy & Security** page that includes:

* A top image (welcome.png) for branding.
* A rounded, light-blue container showing a brief privacy statement.
* Clear communication of data handling: users are informed that their data is private, secure, and never shared.

**7. Contact Us Page**

The **Contact Us** page allows users to:

* View their email (non-editable)
* Enter a phone number and a message
* Submit the message, which is stored in a messages collection in Firestore
* View social icons (Facebook, X/Twitter, and YouTube), which open official websites using url\_launcher

**8. Notifications Page (Designed)**

We designed the **Notifications Page UI**, which includes:

* A switch to enable/disable reminders
* A default reminder time display (08:00 AM)
* A “Save Settings” button  
   This page is purely design-based. The actual logic for notifications will be implemented by another team member using flutter\_local\_notifications.

**Firebase Data Structure**

We used Firestore collections such as:

* /users: stores name, email, age, and createdAt
* /messages: stores submitted contact messages (email, phone, message, timestamp)