İstanbul,2025

“What to wear? ”

“We’ve got you.”

Matchy App

Ulya Büşra PACCİ
(20091010059)

Şule PARTLAK
(20091010042)

Advisor: Dr. Büşra ŞAHİN

https://apps.apple.com/app/matchy-app

Introduction

- **WHAT IS THE PROBLEM ?**

In today’s fast-paced life, people often struggle with choosing what to wear. Our goal is to solve this daily dilemma by offering a digital wardrobe solution that simplifies outfit planning

- **MATCH APP**

The purpose of this project is to develop a smart wardrobe app that helps users manage their clothes and get outfit suggestions based on weather and events. It aims to make dressing easier, more efficient, and personalized by using image uploads and categorized data.

Similiar Apps and Differences

- **AI STYLIST**

Uses AI to assists outfits from user's wardrobe.

It focuses mostly on fashion trends and style aesthetics, while our app also considers weather, event, and location data for functional outfit planning.

- **STYLISTA**

Offers virtual styling tips and outfit inspiration based on occasion and user's style.

More like a fashion social media platform; lacks user wardrobe photo uploads and personalized combination generation based on real-time context.

- **PUREPLE**

Wardrobe organizer with manual outfit planning and sharing combins with people. .

Doesn't use AI or weather/event data; combinations are created manually by the user. Our app automates this process for smart, context-aware suggestions.

Used Technologies

PROGRAMMING LANGUAGES

- **Swift:** Used for developing the iOS app's user interface and handling user interactions.
- **Python:** Used in the backend to process clothing data and generate outfit recommendations.

USED LIBRARIES

- **OpenCV**: Used to read the image and convert the image into RGB color model
- **Scikit-learn:** Used to determine dominant color in clothe image.

DATABASE

- **Firebase:** Used to store user data, clothing information, and retrieve image and text content in real time.

IDES AND TOOLS

- **Xcode:** Used to design and build the iOS app using Swift.
- Visual Studio Code: Used to write and manage Python backend code.
- **Visual Paradigm:** Used for designing UML diagrams like class and activity diagrams.
- **GitHub:** Used for easy accessibility of the written code, as well as to connect with Render then API wors properly
- **Gunicorn:** Used to serve the Flask app as a production-ready WSGI HTTP server.
- **Render:** Used to deploy and host the Python backend and APIs online.

FRAMEWORKS

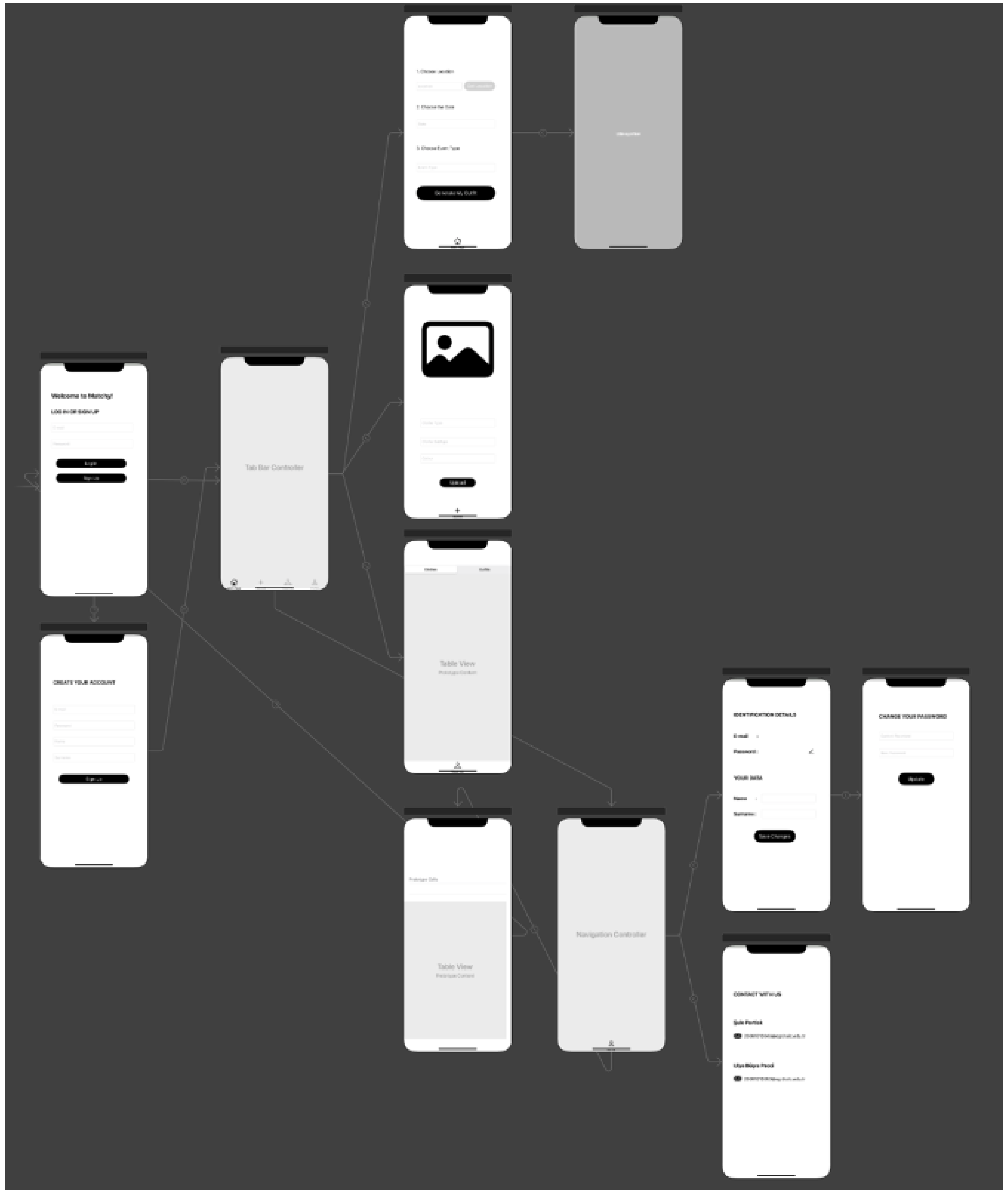
- **UIKit**: Used for building the visual components and layouts of the iOS app.
- **CoreLocation**: Used to fetch the user's current location for weather-based suggestions.
- **Flask**: Used to build the backend API that handles outfit recommendation and clothe analiyses requests.

APIS

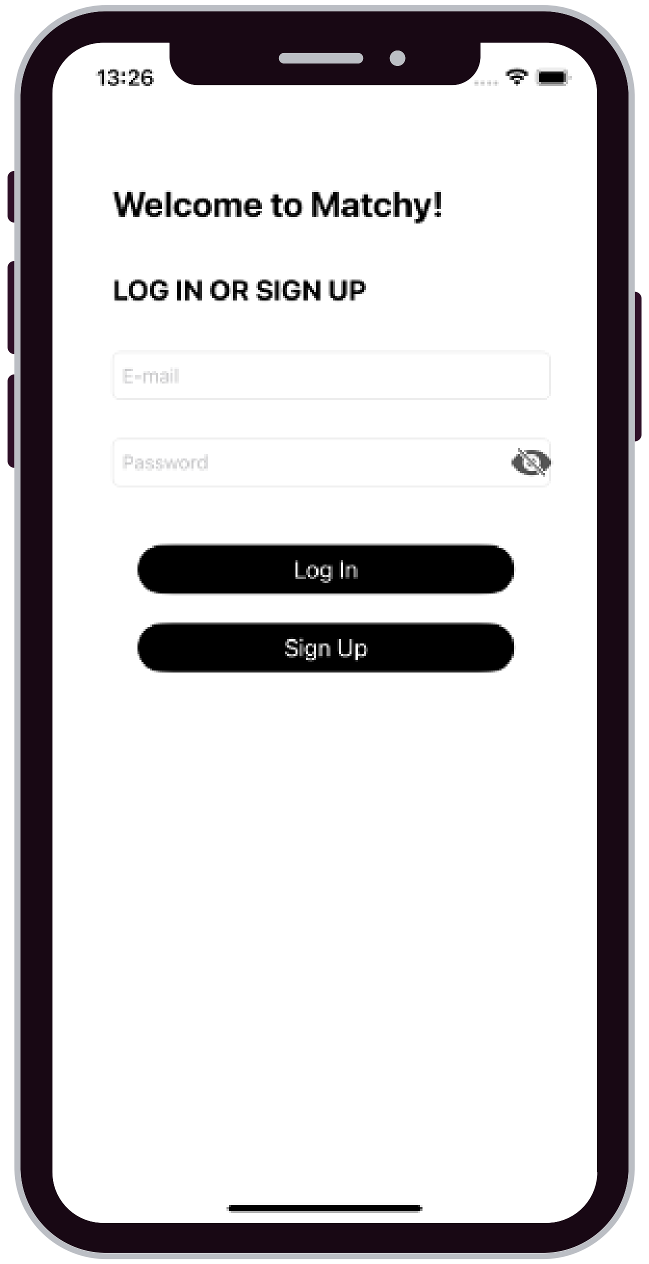
- **Azure Computer Vision API:** Used to analyze uploaded clothing images (e.g. object detection).
- **OpenWeatherMap API:** Used to fetch real-time weather data and suggest suitable outfits.

INTERFACES

- Login Page
- Sing Up Page
- Main Page
- Outfit Page
- Upload Page
- Wardrobe Page
- Profile Page
- Personal Detail Page
- Password Page
- Contact Us Page



LOGIN PAGE

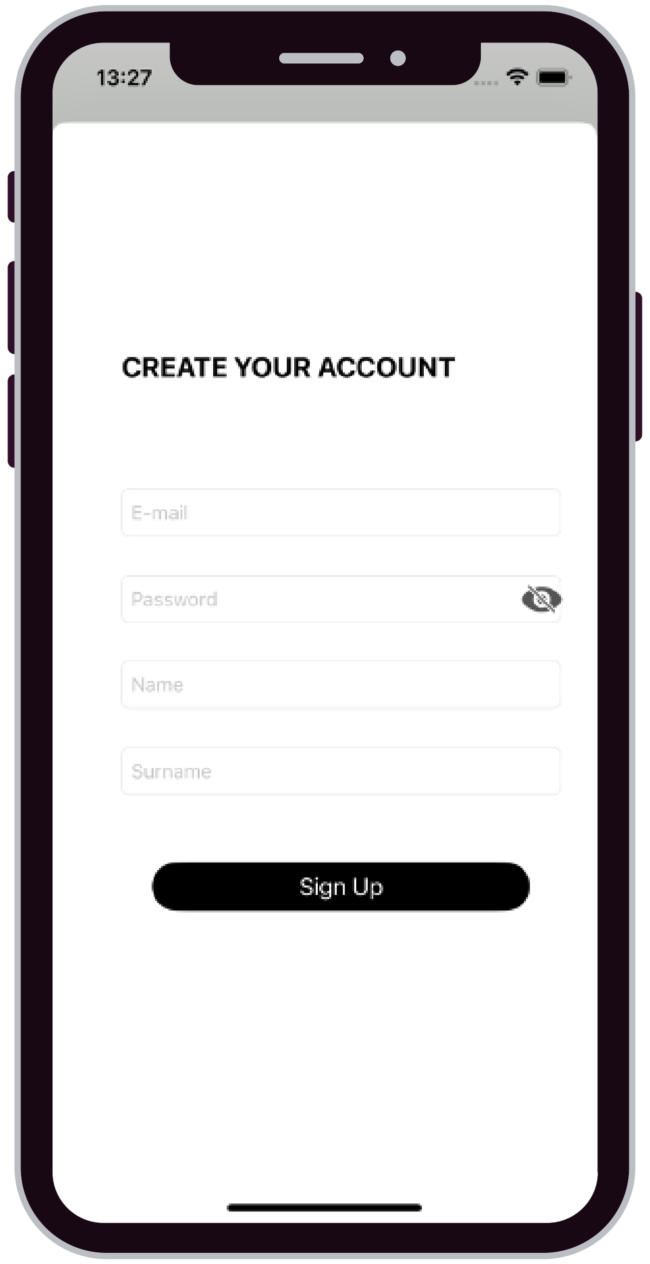


Welcome to Matchy!

LOG IN OR SIGN UP

This is the initial screen of the application where users enter their email and password. If the credentials are correct, the user is directed to the main page. If the user does not have an account, they can go to the sign-up page via a button.

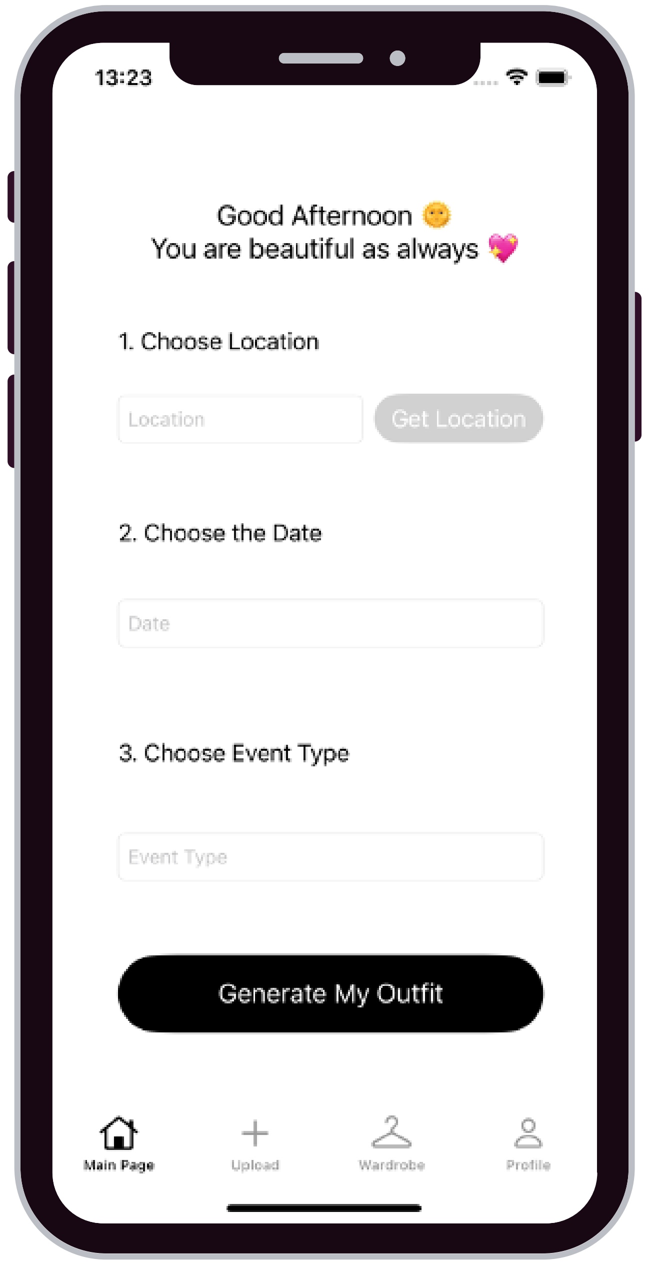
SIGNUP PAGE



CREATE YOUR ACCOUNT

Allows new users to register by entering their name, surname, email, and password. Once completed, the user is directed to the main page and their information is stored in Firebase.

MAIN PAGE



Good Afternoon 😊
You are beautiful as always 💖

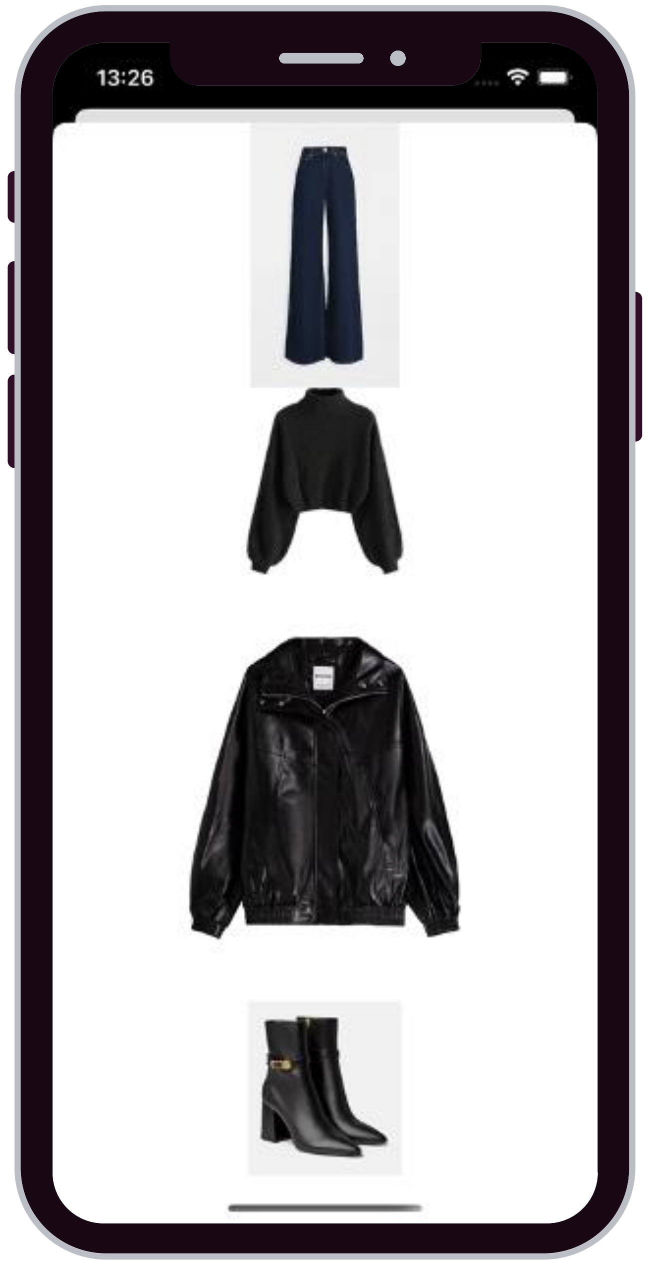
1. Choose Location

2. Choose the Date

3. Choose Event Type

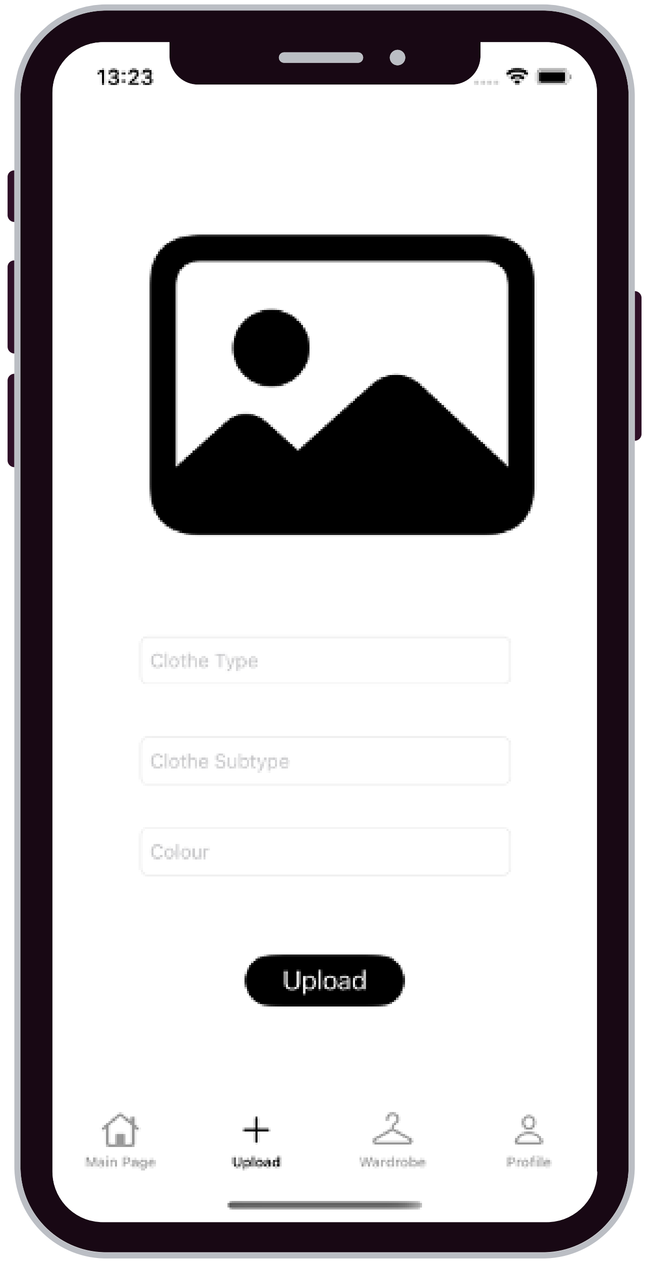
The main screen where users are asked to provide location, date, and event type to receive an outfit suggestion. It includes pickers for date and event, and location can be entered manually or selected using GPS.

OUTFIT PAGE



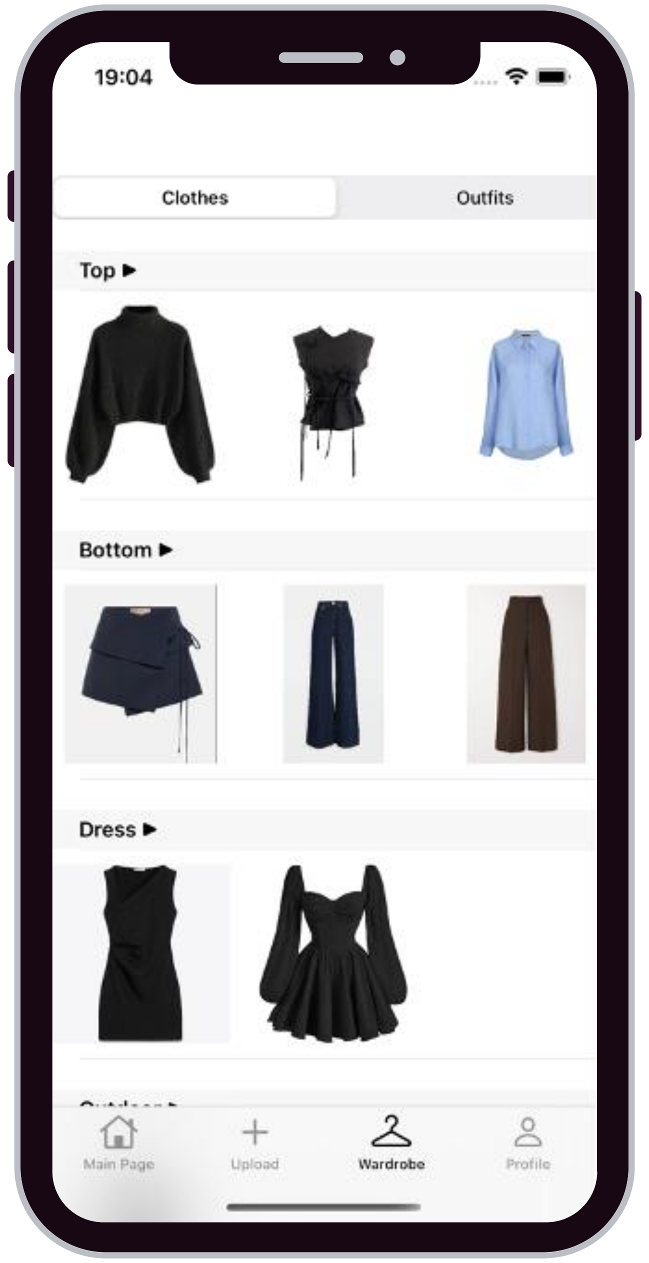
Displays the outfit suggestions generated based on the user's information. Users can swipe right to save a liked outfit or swipe left to request a new one. Saved outfits are stored in the wardrobe.

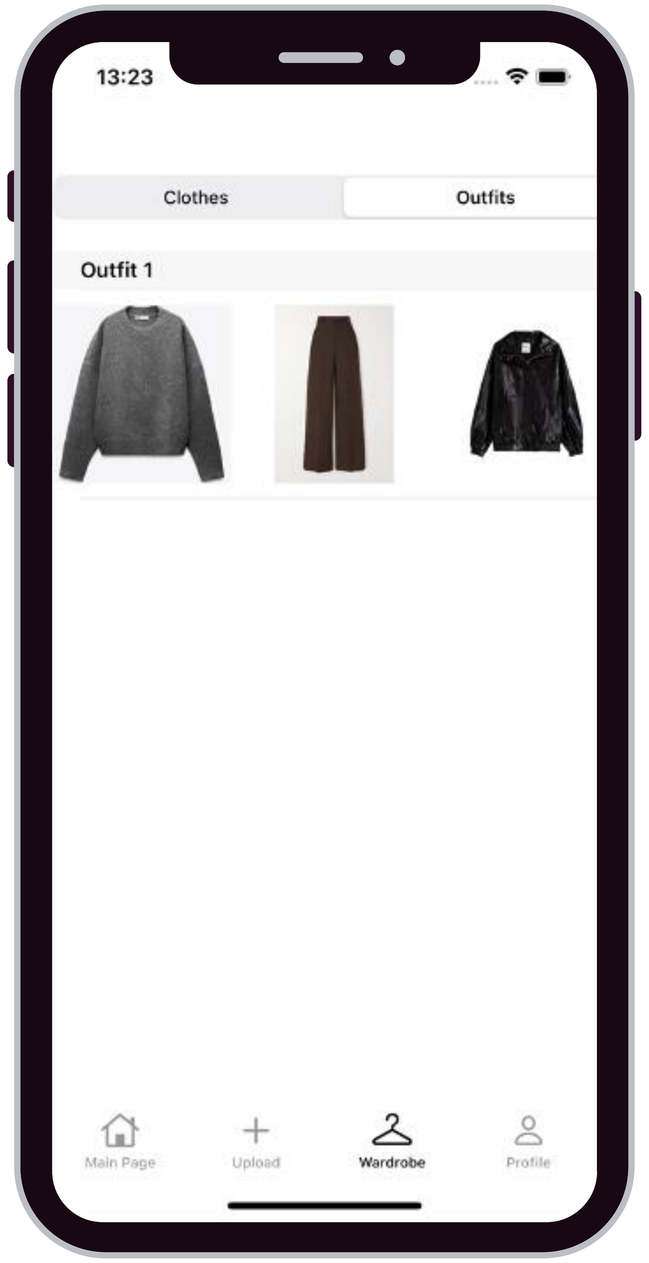
UPLOAD PAGE



Users can upload images of their clothes by choosing from the gallery or using the camera. The system analyzes the image to detect the type, subtype, and color of the clothing automatically. Users can manually edit this data before uploading.

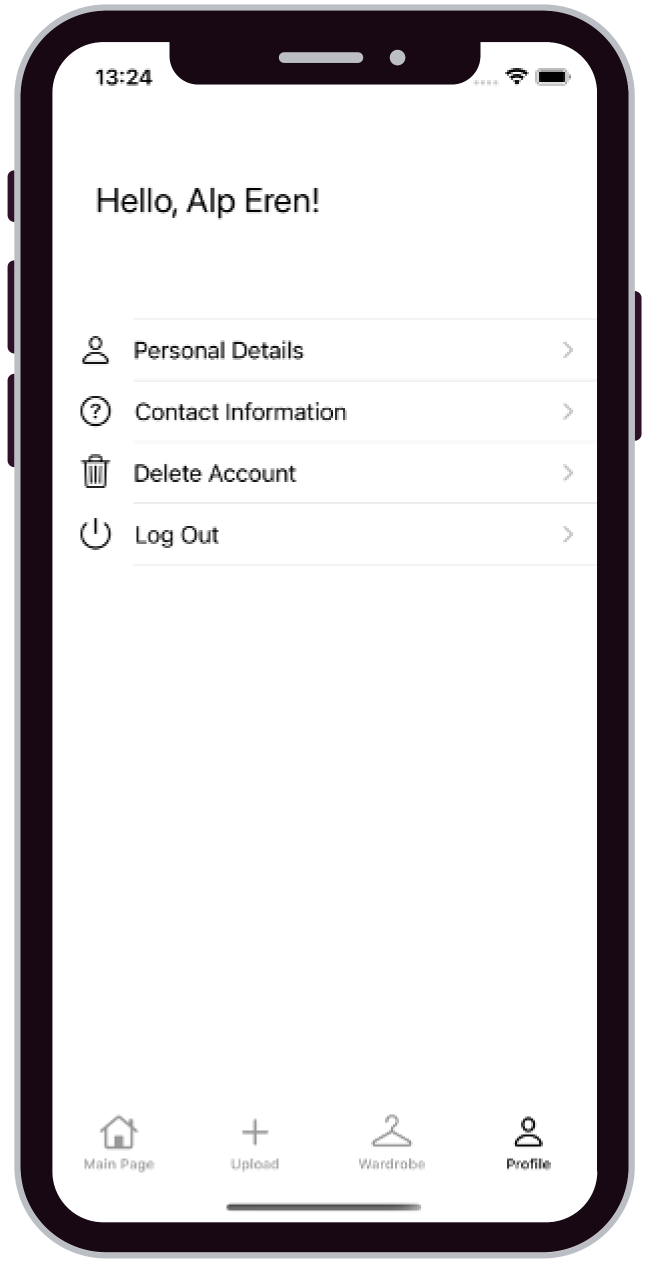
WARDROBE PAGE





Displays all clothes and saved outfits. Clothes are categorized (e.g., Top, Bottom, Shoes). Users can delete items with a long press gesture. The page has two sections: Clothes and Outfits, accessible via a segmented controller.

PROFILE PAGE

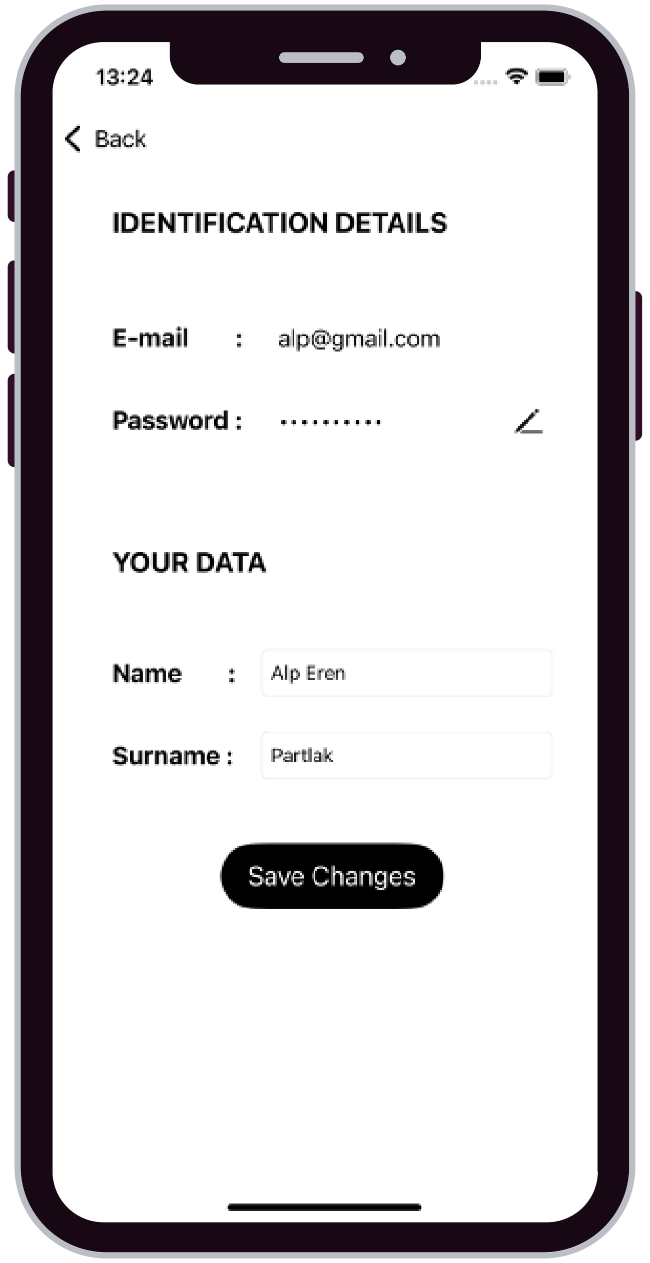


13:24

Hello, Alp Eren!

Shows a greeting with the user's name and includes options like personal details, change password, contact us, delete account, and logout. Each option directs the user to its relevant screen.

PERSONAL DETAIL PAGE



13:24

IDENTIFICATION DETAILS

E-mail : alp@gmail.com

Password : ··········

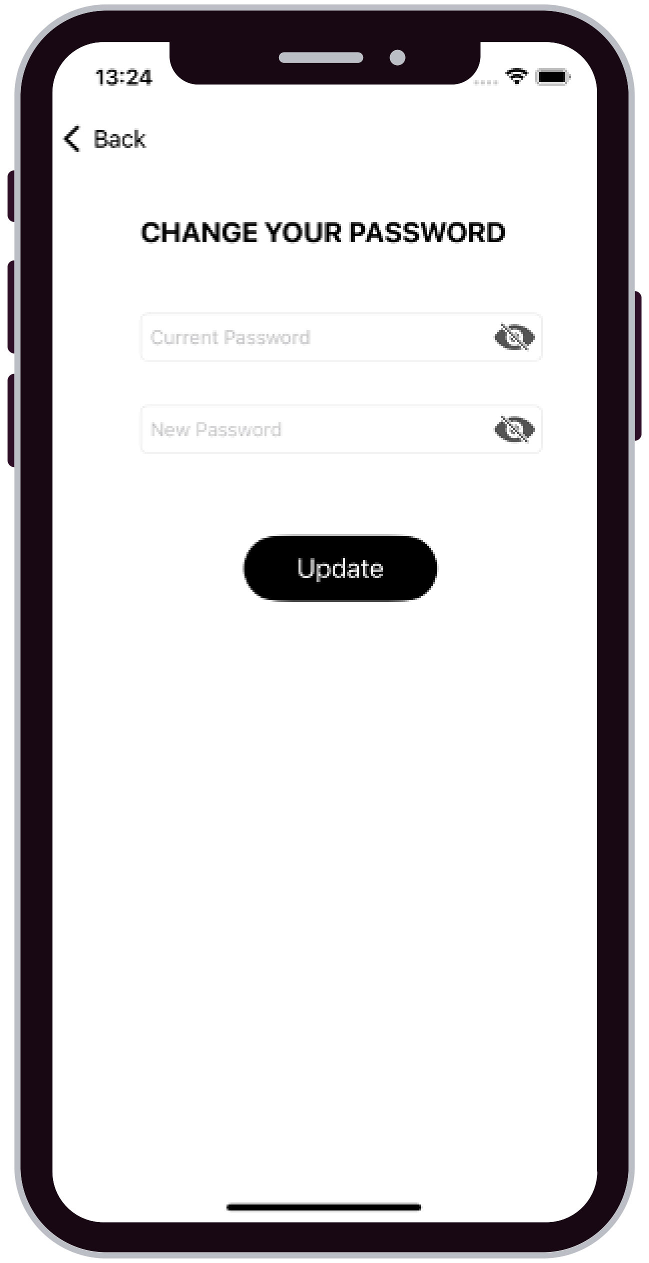
YOUR DATA

Name : Alp Eren

Surname : Partlak

Displays the user's email, and allows the user to update their name and surname. It includes a button to go to the Change Password page.

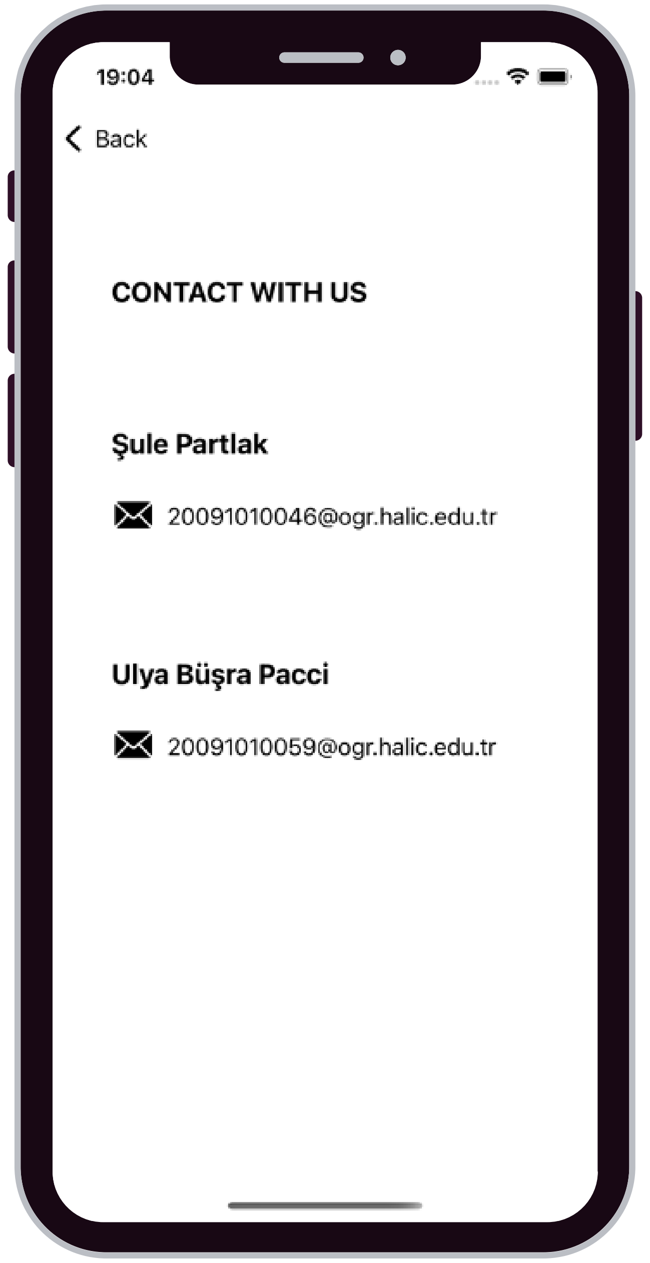
PASSWORD PAGE



CHANGE YOUR PASSWORD

Allows the user to update their password by entering the current and new passwords. The system checks the current password before allowing the change.

CONTACT US PAGE



CONTACT WITH US

Şule Partlak

✉ 20091010046@ogr.halic.edu.tr

Ulya Büşra Pacci

✉ 20091010059@ogr.halic.edu.tr

Shows the developers' full names and email addresses. It's a static informational page where users can see who developed the app.

Algorithms

Outfit Suggestion

This algorithm generates outfit suggestions based on the user's location, date, and event type.

To determine the weather, the system uses OpenWeather API to fetch temperature data for the given date and location. It calculates the average temperature of that day by processing data in 3-hour intervals. Based on the result, the day is classified as cold, mild, or hot.

Then, the system selects suitable clothes from the user’s wardrobe by matching weather, event type, and color compatibility. It starts with a top or dress, adds bottoms, shoes, and outerwear if needed. The final combination is shown to the user.

OpenWeather

Clothe Detection

This algorithm identifies the type and color of a clothing item uploaded by the user.

The uploaded image is analyzed using Azure Computer Vision API, which detects the clothing category based on visual content. Then, the system extracts the dominant color from the image and converts it into a standard CSS3 color name.

Users can review and, if needed, edit the detected information before saving the item to their wardrobe.

Computer Vision

Testing and Encountered Problems

During the development of Matchy App, several issues were encountered and resolved through debugging and user testing.

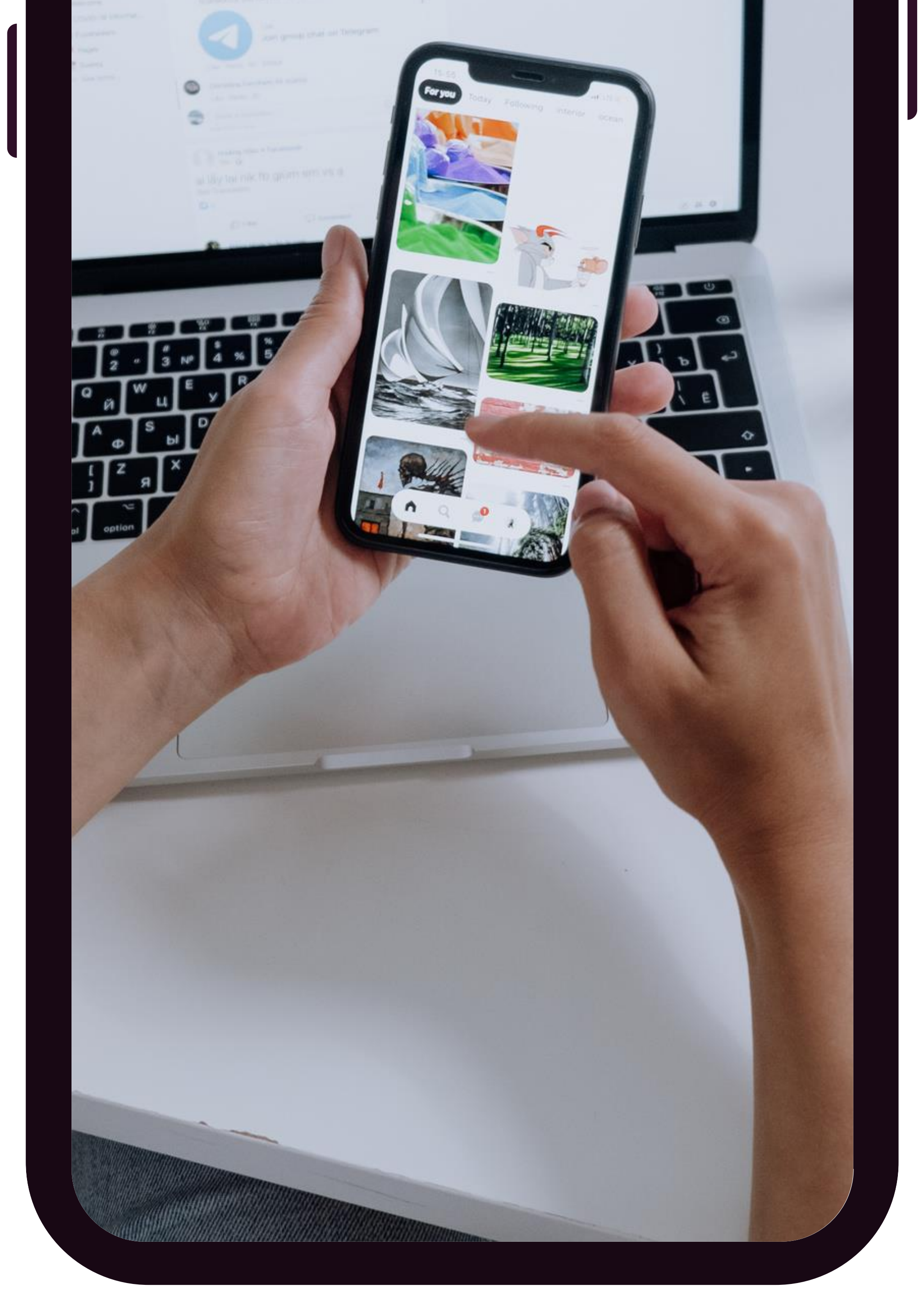
One of the main problems was related to data type mismatches. While sending date information from the iOS interface to the backend API, an error occurred because the date was transmitted as an integer instead of a string. This issue was solved by converting the date format before sending the request.

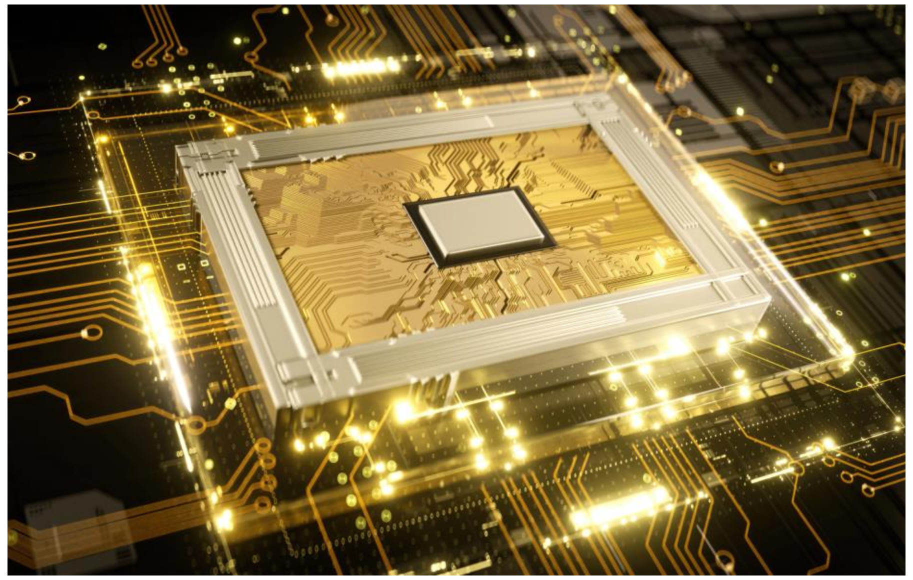
Another challenge was related to Firebase database access restrictions. Since the backend service (Render) used public GitHub links that contained Firebase credentials, access was temporarily blocked by Google for security reasons. The problem was resolved by making the GitHub repository private to protect sensitive data.

Regular testing with different scenarios (e.g., various weather conditions, clothing combinations, and user actions) helped identify logical bugs and improved system stability and performance.

Future Improvements

- A deep learning model can replace third-party tools for better clothing analysis.
- The backend can be optimized for speed and scalability.
- A style learning system could adapt to user preferences.
- An Android version can expand the user base.
- The app can be improved to display outfits directly on the user using virtual try-on technology.



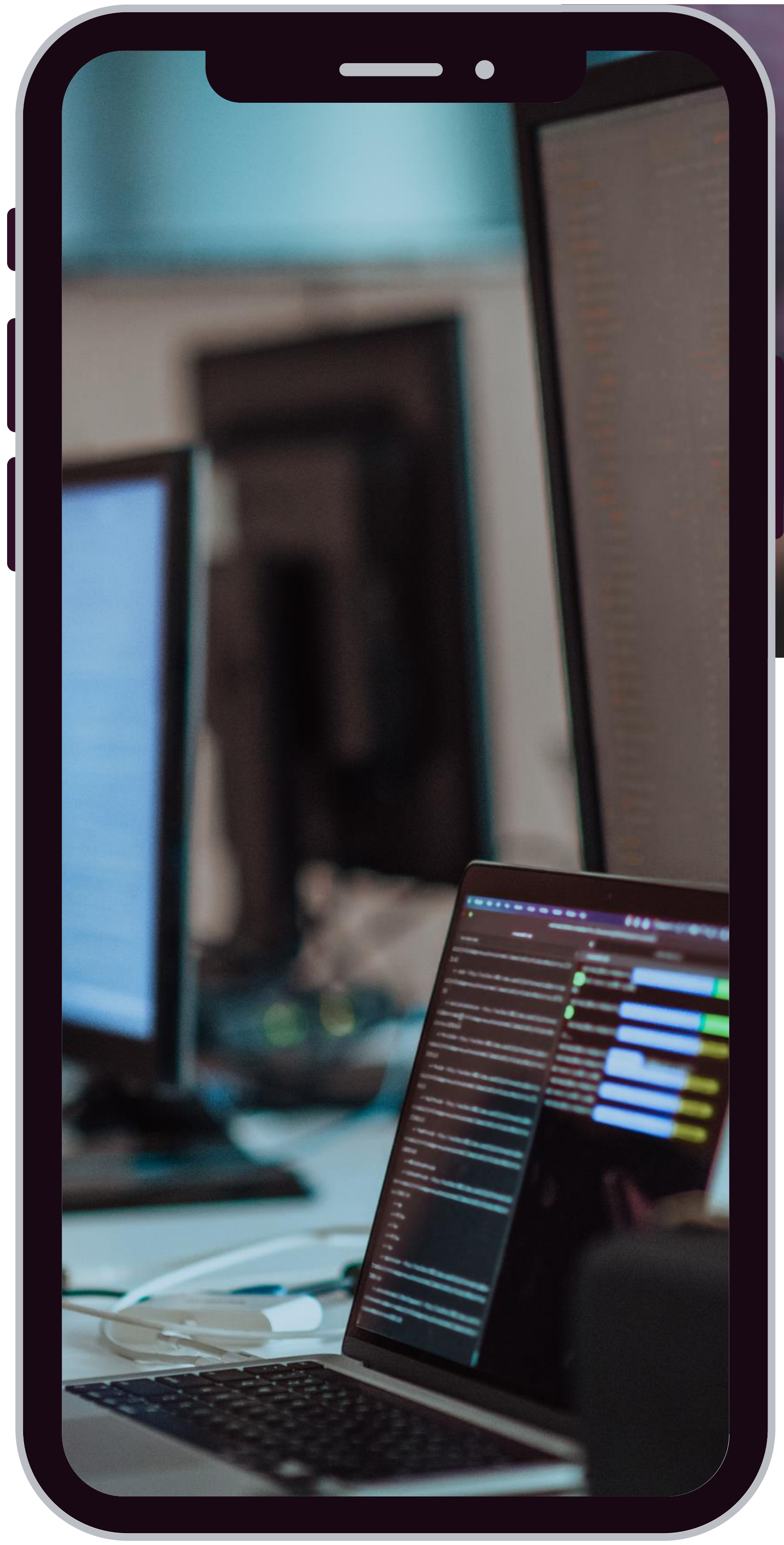


CONCLUSION

Matchy App was developed to offer outfit suggestions based on environmental factors and personal wardrobes, aiming to save users time and promote sustainable fashion choices.

Using AI-powered tools and weather data, the system provides personalized combinations by analyzing the user’s own clothes. Throughout development, modern technologies like Swift, Firebase, and Azure were successfully integrated.

The application achieved its main goal with a clean interface and functional features. Matchy App sets a strong foundation for future enhancements in AI-driven fashion solutions.



Thank You for Your Attention