

CM ASSIGN2

by Liv Collins

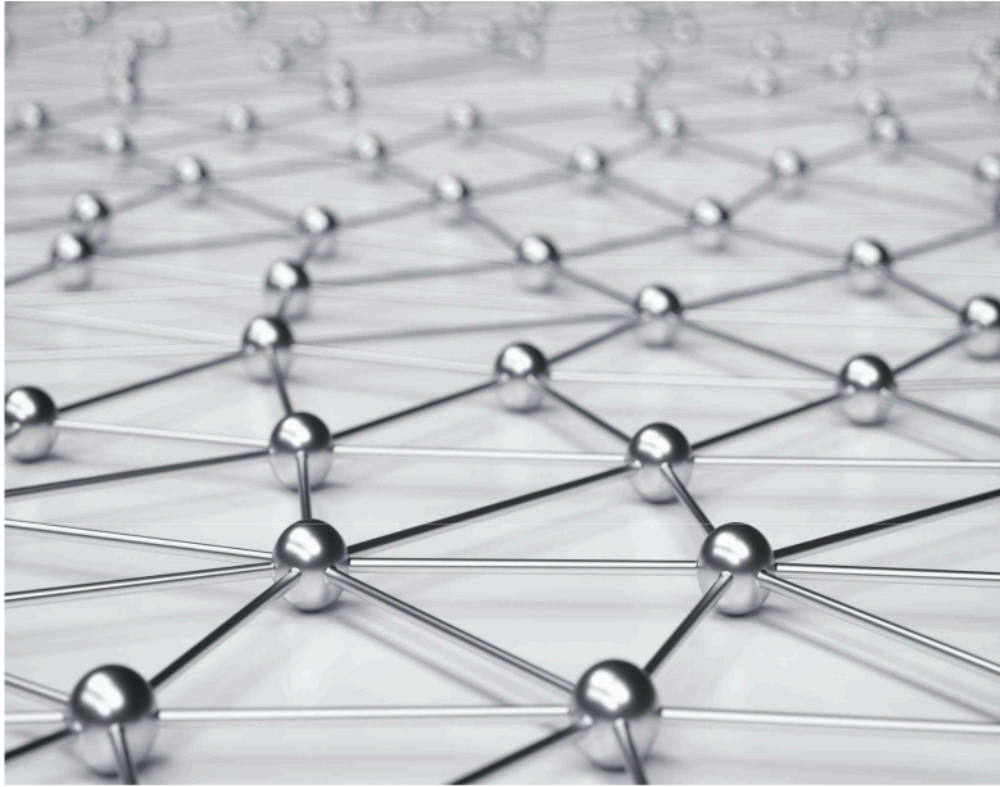
Submission date: 07-May-2026 02:24PM (UTC+0100)

Submission ID: 282620825

File name: 57140_Liv_Collins_CM_ASSIGN2_731414_1029302904.docx (5.03M)

Word count: 3981

Character count: 28662



Assignment 2: Practical & Professional Development

CHANGEMAKERS: CREATIVITY AND VALUE CREATION

Olivia Collins 2403891 | ACCA5033 | 07/05/2026 | Word Count: 2136

Table of Contents

Table of Contents.....	1
Introduction	1
Task 1: Practical Development/ Implementation	2
Q.1 Initial Feedback.....	2
Data Collection Method	2
Summary of Feedback	2
Changes Made Based on Feedback.....	4
Q.2 Implementation.....	4
Overview of Implementation.....	4
Task 2: Enterprise.....	11
Task 3: Project Feedback & Reflection	14

Introduction

Selecting appropriate clothing for changing weather conditions can be inconvenient and time-consuming, especially when factors such as temperature, rainfall, and wind must all be considered. Many people check weather forecasts but still struggle to decide what to wear. The Wear the Weather project aims to solve this problem by developing an application that provides simple and practical outfit recommendations based on real-time weather conditions.

The application uses location-based weather data to generate suggestions that improve user comfort and decision-making. It is designed with a focus on usability, allowing users to quickly receive outfit advice tailored to current environmental conditions.

This report presents the development and evaluation of the project. It includes the design and implementation process, user feedback collected during development, and reflections on the overall project progress. In addition, the report explores enterprise opportunities, including business strategies and external factors that may influence its success.

Task 1: Practical Development/ Implementation

Q.1 INITIAL FEEDBACK

Data Collection Method

To evaluate the Wear the Weather application idea, a small-scale data collection process was conducted to gather initial user feedback. A questionnaire method was used through the online medium of Microsoft Forms. The questionnaire was anonymous for privacy reasons, but all participants consented to their responses being used within this report.

A total of eight participants, aged between 18 and 25, were selected as they represent the target audience for a mobile-based lifestyle application. The questionnaire included a combination of multiple choice and open-ended questions designed to assess user interest, usability expectations, and potential concerns.

Some of the questions include:

- How useful do you think an app that suggests an outfit based on the weather would be?
- How often do you check the weather before picking outfits in the morning?
- How likely are you to download this app if it were available?

Summary of Feedback

The feedback collected was largely positive, with all participants expressing interest in the concept of the application. Some users answered the question “What do you like most about the idea of the Wear the Weather app?” These were the answers provided:

8. What do you like most about the idea of the Wear the Weather app?

4 Responses

ID ↑	Name	Responses
1	anonymous	Being able to tailor my outfit to the weather means im prepared for any temperature changes or possible rainfall.
2	anonymous	That it would provide useful insight to help you plan your outfit for the day
3	anonymous	Clothing suggestions are an incentive for looking at the weather it's cool
4	anonymous	Could remind you to take an umbrella

These answers provided a positive overview of the idea; this helps with what doesn't need to change and where the strong points are.

Several common themes emerged during the questionnaire:

1. Usefulness and Convivence

Participants highlighted that the application would save time and reduce the effort required to decide what to wear each day. More participants check the weather.

2. How often do you check the weather before picking your outfits in the morning?

[More details](#)



2. Personalisation

Users suggested that the application should include options to customize recommendations based on personal style preferences. With some participants adding these suggestions:

5. Are there any additional features that you would like to see in the app?

2 Responses

ID ↑	Name	Responses
1	anonymous	Perhaps small suggestions based on preferences that suggest the user clothing items that they could buy from different websites
2	anonymous	Fund the app through advertising with clothing stores or listings

Adding adverts and links to external websites, users can find clothes they like that suits the climate that they live in.

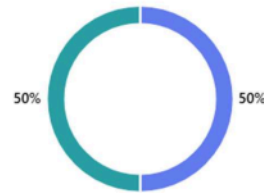
3. Visual Features

The participants were torn between the want for an avatar or not. With a 50/50 spilt between yes and maybe.

7. Would it be helpful to see outfit suggestions displayed on a customizable avatar?

[More details](#)

● Yes	4
● No	0
● Maybe	4



This helped decide to make this feature optional, so users can pick what interface they would like to choose.

(Full questionnaire results can be found in the Appendix)

Changes Made Based on Feedback

Based on the feedback received, several improvements were identified and incorporated into the project design:

- A basic personalisation feature was introduced, allowing users to select preferred clothing styles.
- The system logic was refined to improve the accuracy of outfit recommendations based on multiple weather conditions.
- Plans were made to include visual outfit suggestions in the future development stages.

These changes ensured that the application better aligns with user expectation and enhances overall usability.

Q.2 IMPLEMENTATION

Overview of Implementation

The Wear the Weather application was developed as a prototype to demonstrate how weather data can be used to generate outfit recommendations. The implementation focused on integrating weather data, processing inputs and producing meaningful outputs for the user.

Tools and Technologies Used

The development of the application involved the use of several tools and technologies.

The hardware used for this project was an MSI laptop with Intel Core i7 and an Nvidia GeForce RTX 4060. For the software, Windows Command Prompt was used alongside Expo and Native React. The coding was all completed in Visual Studio Code using JavaScript as the coding language. The phone that was used to conduct the tests was an iPhone 16 using the Expo Go app. The API that was used to collect real-time weather data was OpenWeather API. [1]

Prior to this project, I did not have any knowledge of how to use Expo and JavaScript. So, I used LinkedIn Learning to help broaden my knowledge and overall skill set.

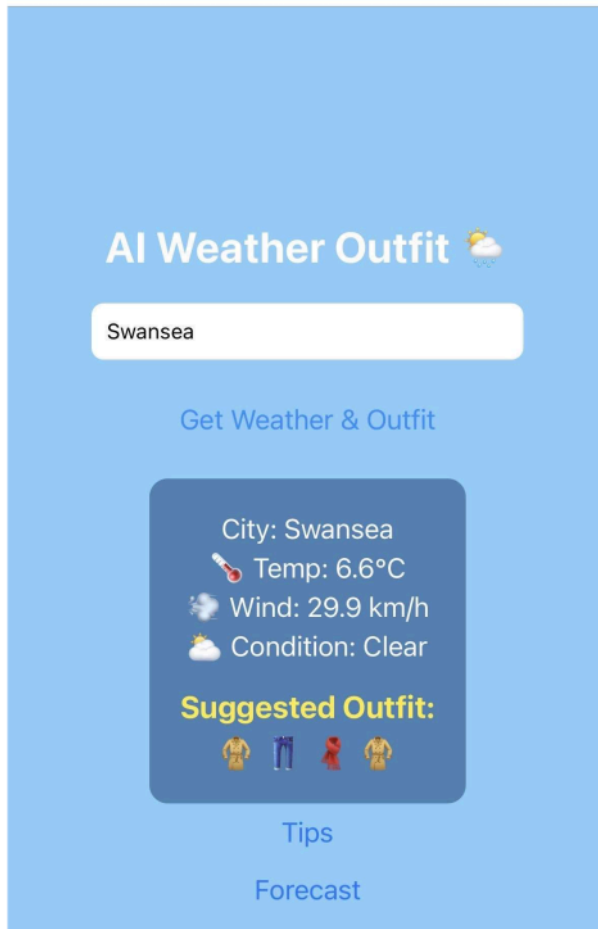
(Proof in Appendix)

User Interface Development

The first stage of implementation focused on creating a simple and user-friendly interface. The interface allows the user to enter a location and receive clothing recommendations based on the weather conditions in that area. Simplicity was prioritised to ensure that the application was easy to use.

Main Application Interface:

Home



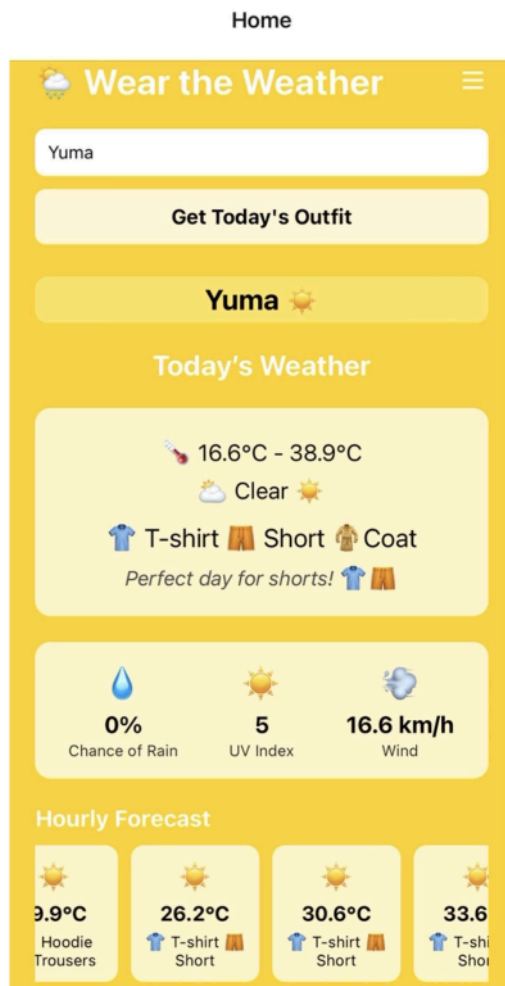
This was the first iteration of the home screen, just to make sure that the Api and the code work in the correct way.

The most recent version of the home screen is a lot more detailed with animation and an hourly forecast.

Home



And for warmer conditions, the colour scheme changes to match:



Weather API Integration

To generate accurate recommendations, the application required real-time weather data. A weather API was integrated into the project to retrieve information such as temperature, rainfall, and wind speed. The API connection was implemented using HTTP requests within the application code.

The code:

```
// Weather request
const weatherRes = await axios.get(
```

```
`https://api.open-  
meteo.com/v1/forecast?latitude=${latitude}&longitude=${longitude}&hourly=te  
mperature_2m,weathercode,windspeed_10m,precipitation_probability&daily=te  
mperature_2m_max,temperature_2m_min,weathercode,windspeed_10m_max,precipit  
ation_probability_max,uv_index_max&timezone=auto`  
);
```

```
// Get coordinates  
const geoRes = await axios.get(  
  `https://geocoding-api.open-meteo.com/v1/search?name=${city}`  
);
```

These demonstrate successful retrieval of the weather information and location from the API. The returned data includes temperature values, weather conditions and wind speed measurements,

Recommendation Logic Implementation

The core functionality of the application involved generating outfit recommendations based on the weather data received from the API. A rule-based decision system was implemented using conditional statements.

For example:

- Cold weather resulted in recommendations for coats and warm clothing
- Rain triggered an umbrella option
- Heavy rain prompts an umbrella and coat.
- High wind speeds prompted layered clothing recommendations
- Snow prompts winter gloves

```
const suggestOutfitEmojis = (temp, wind, condition) => {  
  let outfit = '';  
  if (temp > 25) outfit += '👕👖';  
  else if (temp >= 15) outfit += '👕🧥';  
  else outfit += '🧥🧣🧤';  
  if (wind > 10) outfit += ' 🧥';  
  if (['Drizzle', 'Light Rain', 'Showers'].includes(condition)) outfit += '  
  🌧️';
```

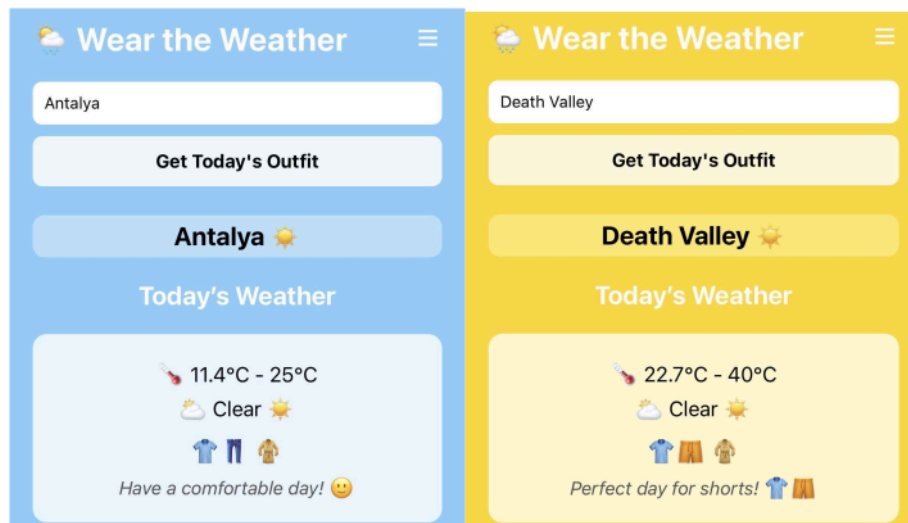
```

if (['Heavy Rain', 'Heavy Showers', 'Thunderstorm'].includes(condition))
outfit += ' 🌧️👕';
if (condition === 'Snow') outfit += ' ❄️🧤';
return outfit;
};

```

(Refer to Appendix for full code implementation)

This is an example of the outfit recommendation logic with different weather conditions:



Initially, the recommendation system only did rain or sun. Which did not produce the most accurate outfit recommendations.

Output Testing

After implementing the recommendation system, testing was carried out using multiple different weather scenarios to verify that the outputs matched expected conditions. Over 25 different locations around the world were tested to ensure the recommendations remained consistent and logical.



Overall, the tests have raised confidence in the ability of coding in JavaScript since it was a new skill that had to be learned for this assignment.

Challenges and Troubleshooting

API Connectivity Issues

API requests sometimes did not work and returned with errors due to invalid request formatting. This was solved by reviewing the API documentation and correcting the request syntax.

User Interface Problems

Early versions of the interface appeared cluttered on different screens. The layout was redesigned to improve spacing and responsiveness.

Recommendation Accuracy

The initial logic system only considered temperature and precipitation. The recommendation process was improved by incorporating wind speed into the decision-making process.

Implementation Summary

The implementation process successfully produced a functional prototype capable of generating weather-based outfit recommendations. The combination of live weather data and decision-making logic showed how digital technologies can improve everyday life.

Task 2: Enterprise

Q.1 BUSINESS OVERVIEW

Start-up Name

The proposed tech start-up name for the Wear the Weather application would be WearWise AI. The name reflects the core purposed of the product which is to provide AI powered clothing recommendations based on the weather. Alongside that, WearWise AI could be used for other products, whether that be a digital wardrobe or a shopping app connected to other WearWise applications'.

Logo Concept

The logo design would combine a cloud icon with a t-shirt (common clothing item) to visually connect the weather forecast side of the app with the clothing side. The design would use a clean, minimal style to appeal to a modern mobile application audience. So, the logo would stand out, but not in a negative way

Proposed idea for the design:



(Mock-up made on Canva)

Business Type

The business would operate as a Private Limited Company (Ltd). This structure is suitable because it limits personal liability for the owners while allowing investment opportunities and growth.

Business Strategy

The business strategy for WearWise AI would focus on a “freenium” mobile application model:

The free version would provide the user with basic outfit recommendations based on the weather and limited personalisation options.

The premium subscription would allow full access to advanced AI-Based outfit personalisation, style preferences, an outfit calendar and an ad-free experience.

The premium subscriptions and advertisements would allow the right funding needed.

The target audience for Wear the Weather are students, workers and commuters who need a quick way to decide on their daily outfits without fuss.

The marketing strategy would rely heavily on social media advertising (TikTok, Instagram), Influencer partnerships in fashion and lifestyle and App Store optimisation and advertisement.

Q.2 PEST ANALYSIS

Political

WearWise AI must comply with data protection regulations such as GDPR, as the application may collect location data. Government policies around digital privacy and AI could impact how user data is stored and processed.[3]

Economic

The AI and mobile application market is experiencing rapid growth alongside the spike in numbers of mobile phone users, [6] which creates a positive opportunity for the business. However, economic factors such as inflation and reduced consumer spending could affect users' willingness to pay for additional subscriptions. Although, being the first app like Wear the Weather, users may feel inclined to pay the extra fees just to experience the app to its fullest.

Social

Modern users increasingly rely on smartphone for daily tasks more than ever, including checking the weather and lifestyle planning. There is also a growing demand for convenience-based applications that reduce time and effort in daily routines. However, some users may have concerns about privacy and how their data location is used.

Technological

Advancements in AI and machine learning directly supports the development of WearWise AI. [4] [5] Reliable weather APIs such as OpenWeather enable accurate data retrieval. However, technological dependency introduces new risks regarding API downtime and possible update wait times.

Legal and Ethical Considerations

The application must ensure:

- Full compliance with data protection laws
- Transparent data usage policies
- Secure storage of user information
- Ethical AI usage, avoiding biased or inaccurate recommendations

Task 3: Project Feedback & Reflection

Q.1 PRACTICAL FEEDBACK

Feedback on the final implementation of the Wear the Weather application was collected through face-to-face interviews after four users tried the app themselves. Users tested the application by entering locations and reviewing the outfit recommendations generated.

The feedback was mostly positive, with participants stating that the application was easy to use and useful for daily outfit decisions. Users particularly liked the clear interface and quick recommendations.

Q.2 PROJECT PROGRESS DIARY

The progress diary helped with keeping time in check and progress constant. Every week a weekly progress sheet was filled out with the current progress and the actions that needed to be completed.

Q.3 SKILLS

The project helped develop both technical and transferable skills. Technical skills included programming, API integration and interface design. Problem solving skills were also improved on during this project due to the number of problems that raised during coding.

Q.4 POSITIVE SOCIAL IMPACT

The Wear the Weather app could positively impact society by helping users make quicker and more practical outfit decisions during changing weather conditions. The application will reduce time and effort spent for people with busy lives, helping them maintain a smooth and streamline daily routine.

Conclusion

Overall, this project has been difficult but very rewarding. Many challenges and issues were overcome using new and developed skills. Hopefully, this project will become a fully developed application that can help people with their daily routines.

This report describes the implementation progress, the different stages of ideas and overall Wear the Weather development.

References

- [1] OpenWeather, "Current Weather Data API," OpenWeather, 2026. [Online]. Available: <https://openweathermap.org/api>. [Accessed: 7-May-2026].

- [2] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Harlow, U.K.: Pearson, 2021.

- [3] UK Government, "Guide to the General Data Protection Regulation (GDPR)," GOV.UK, 2026. [Online]. Available: <https://www.gov.uk/data-protection>. [Accessed: 7-May-2026].

- [4] Microsoft, "What is Machine Learning?," Microsoft Azure, 2025. [Online]. Available: <https://azure.microsoft.com/en-gb/resources/cloud-computing-dictionary/what-is-machine-learning/>. [Accessed: 7-May-2026].

- [5] IBM, "What is Artificial Intelligence (AI)?," IBM, 2025. [Online]. Available: <https://www.ibm.com/topics/artificial-intelligence>. [Accessed: 7-May-2026].

- [6] Statista, "Artificial Intelligence (AI) Market Size Worldwide," Statista, 2025. [Online]. Available: <https://www.statista.com/topics/3104/artificial-intelligence-ai-worldwide/>. [Accessed: 7-May-2026].

- [7] Visual Studio Code, "Visual Studio Code Documentation," Microsoft, 2026. [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed: 7-May-2026].

[8] Python Software Foundation, "Python Documentation," Python.org, 2026. [Online]. Available: <https://docs.python.org/3/>. [Accessed: 7-May-2026].

[9] Google Developers, "Web API Design Best Practices," Google for Developers, 2025. [Online]. Available: <https://developers.google.com>. [Accessed: 7-May-2026].

[10] Kickstarter, "Crowdfunding Creative Projects," Kickstarter, 2026. [Online]. Available: <https://www.kickstarter.com>. [Accessed: 7-May-2026].

[11] Investopedia, "PEST Analysis," Investopedia, 2025. [Online]. Available: <https://www.investopedia.com/terms/p/pest-analysis.asp>. [Accessed: 7-May-2026].

[12] LinkedIn Learning, "Programming and Software Development Courses," LinkedIn Learning, 2026. [Online]. Available: <https://www.linkedin.com/learning/>. [Accessed: 7-May-2026].

Appendix

Project progress form

Name: Olivia Collins (2403891)

Programme: Applied Computing

Project Aim: To develop Wear the Weather, an application that provides outfit recommendations based on weather conditions such as temperature, rainfall, and wind speed to improve user convenience and daily decision-making.

Project Objectives:

1. Research and analyse existing weather and clothing recommendation applications.
2. Design and develop a functional prototype capable of generating outfit recommendations.

3. Collect user feedback to evaluate and improve the application design and functionality.

Weekly Project progress forms:

Week 1

Date: 11/03/2026

Progress:

Initial project planning and research were completed. Existing weather applications and outfit recommendation systems were analysed to identify useful features and possible improvements. Early ideas for the application interface and functionality were also discussed.

Actions:

State priority actions to undertake over the next week below.

- Begin researching weather APIs
- Create initial design ideas and wireframes
- Plan implementation stages for development

Next review date: 18/03/2026.

Week 2

Date: 18/03/2026

Progress:

Research into weather APIs was completed and OpenWeather API was selected for the project. Basic wireframes for the application interface were created to plan the layout and user interaction.

Actions:

- Set up development environment in Visual Studio Code
- Begin coding the basic user interface
- Test API connection

Next review date: 25/03/2026

Week 3

Date: 25/03/2026

Progress:

The development environment was successfully configured, and the basic structure of the application was created. Initial API testing was completed, and weather data could be retrieved successfully.

Actions:

- Improve interface layout and styling
- Begin implementing recommendation logic
- Test API outputs with different locations

Next review date: 01/04/2026

Week 4

Date: 01/04/2026

Progress:

The outfit recommendation logic was implemented using conditional statements based on temperature and rainfall. The interface design was improved to make the application easier to navigate.

Actions:

- Add wind speed functionality
- Improve recommendation accuracy
- Continue interface testing

Next review date: 08/04/2026

Week 5

Date: 08/04/2026

Progress:

Wind speed conditions were successfully added to the recommendation system. Multiple weather scenarios were tested to ensure recommendations changed dynamically based on conditions.

Actions:

Conduct initial user feedback survey
Identify improvements from testing
Fix any interface issues discovered during testing

Next review date: 15/04/2026

Week 6

Date: 15/04/2026

Progress:

User feedback was collected using questionnaires and informal discussions. Participants suggested adding personalisation options and visual outfit examples. Minor user interface adjustments were also completed.

Actions:

- Refine recommendation logic
- Improve readability of outputs
- Begin documenting implementation process

Next review date: 22/04/2026

Week 7

Date: 22/04/2026

Progress:

The recommendation system was refined to improve the accuracy of clothing suggestions. Additional testing was completed using different weather conditions and locations.

Actions:

- Capture screenshots for report evidence
- Organise implementation documentation
- Continue testing system stability

Next review date: 29/04/2026

Week 8

Date: 29/04/2026

Progress:

Screenshots of the application interface, API integration, and recommendation outputs were captured for inclusion in the report. Work also began on writing the implementation and reflection sections.

Actions:

- Complete Task 1 report section
- Begin Task 2 enterprise section
- Review references and citations

Next review date: 06/05/2026

Week 9

Date: 06/05/2026

Progress:

The report sections were reviewed and updated. Final testing of the application prototype was completed and remaining formatting improvements were made to the documentation.

Actions:

- Finalise conclusion and references
- Proofread report for errors
- Prepare final submission to Turnitin

Next review date: 13/05/2026

```
Microsoft Windows [Version 10.0.26200.8246]
(c) Microsoft Corporation. All rights reserved.

C:\Users\livvy>cd MyNewApp

C:\Users\livvy\MyNewApp>npx expo start|
```

CODE IMPLEMENTATION:

Home Screen

```
import React, { useState, useEffect, useRef } from 'react';
import { View, Text, TextInput, StyleSheet, TouchableOpacity, Modal,
Pressable, Animated, Dimensions, ScrollView } from 'react-native';
import { useNavigation } from '@react-navigation/native';
import axios from 'axios';

const { width, height } = Dimensions.get('window');

// --- Weather Helpers ---
const getWeatherCondition = (code) => {
  if ([0,1,2].includes(code)) return 'Clear';
  if ([3,45,48].includes(code)) return 'Cloudy';
  if ([51,53,55].includes(code)) return 'Drizzle';
  if ([61,63].includes(code)) return 'Light Rain';
  if ([65].includes(code)) return 'Heavy Rain';
  if ([71,73,75].includes(code)) return 'Snow';
  if ([80,81].includes(code)) return 'Showers';
  if ([82].includes(code)) return 'Heavy Showers';
  if ([95,96,99].includes(code)) return 'Thunderstorm';
  return 'Unknown';
};

const getWeatherEmoji = (condition) => {
  switch (condition) {
    case 'Clear': return '☀️';
    case 'Cloudy': return '☁️';
    case 'Drizzle':
    case 'Light Rain':
```

```

    case 'Showers': return '🌧️';
    case 'Heavy Rain':
    case 'Heavy Showers':
    case 'Thunderstorm': return '⚡️';
    case 'Snow': return '❄️';
    default: return '';
  }
};

const suggestOutfitEmojis = (temp, wind, condition) => {
  let outfit = '';
  if (temp > 25) outfit += '👕👖';
  else if (temp >= 15) outfit += '👕👖';
  else outfit += '🧥👖';
  if (wind > 10) outfit += '🧥';
  if (['Drizzle', 'Light Rain', 'Showers'].includes(condition)) outfit += '☔️';
  if (['Heavy Rain', 'Heavy Showers', 'Thunderstorm'].includes(condition))
  outfit += '☔️🧥';
  if (condition === 'Snow') outfit += '❄️🧤';
  return outfit;
};

// --- Falling Emoji Component
const FallingEmoji = ({ emoji }) => {
  const translateY = useRef(new Animated.Value(-50)).current;
  useEffect(() => {
    const loopAnimation = () => {
      translateY.setValue(-50);
      Animated.timing(translateY, {
        toValue: height + 50,
        duration: 4000 + Math.random() * 2000,
        useNativeDriver: true,
      }).start(() => loopAnimation());
    };
    loopAnimation();
  }, []);
  const left = Math.random() * (width - 30);
  return (
    <Animated.Text
      style={{
        position: 'absolute',

```

```

        left,
        fontSize: 24,
        transform: [{ translateY }]
      }}
    >
      {emoji}
    </Animated.Text>
  );
};

export default function HomeScreen({ city, setCity }) {
  const navigation = useNavigation();
  const [menuVisible, setMenuVisible] = useState(false);
  const [inputCity, setInputCity] = useState(city);
  const [todayWeather, setTodayWeather] = useState(null);
  const [hourlyForecast, setHourlyForecast] = useState([]);
  const [loading, setLoading] = useState(false);
  const [tip, setTip] = useState('');

  // Animation refs
  const statAnim = useRef(new Animated.Value(0)).current;
  const hourAnims = useRef([]).current;

  const getTodayWeather = async () => {
    if (!city) {
      alert('Enter a city name');
      return;
    }

    setLoading(true);

    try {
      // Get coordinates
      const geoRes = await axios.get(
        `https://geocoding-api.open-meteo.com/v1/search?name=${city}`
      );

      if (!geoRes.data.results || geoRes.data.results.length === 0) {
        alert('City not found');
        setLoading(false);
        return;
      }
    }
  }
}

```

```

const { latitude, longitude } = geoRes.data.results[0];

// Weather request
const weatherRes = await axios.get(
  `https://api.open-
meteo.com/v1/forecast?latitude=${latitude}&longitude=${longitude}&hourly=te
mperature_2m,weathercode,windspeed_10m,precipitation_probability&daily=te
mperature_2m_max,temperature_2m_min,weathercode,windspeed_10m_max,precipit
ation_probability_max,uv_index_max&timezone=auto`
);

const daily = weatherRes.data.daily;
const hourly = weatherRes.data.hourly;

const {
  temperature_2m_max,
  temperature_2m_min,
  weathercode,
  windspeed_10m_max,
  precipitation_probability_max,
  uv_index_max,
} = daily;

const condition = getWeatherCondition(weathercode[0]);

const outfit = suggestOutfitEmojis(
  temperature_2m_max[0],
  windspeed_10m_max[0],
  condition
);

// Main weather
setTodayWeather({
  tempMax: temperature_2m_max[0],
  tempMin: temperature_2m_min[0],
  wind: windspeed_10m_max[0],
  condition,
  outfit,
  rainChance: precipitation_probability_max[0],
  uv: uv_index_max[0],
});

```

```

// Hourly forecast
const hourlyData = Array.from({ length: 12 }).map((_, i) => ({
  temp: hourly.temperature_2m[i],
  code: hourly.weathercode[i],
  wind: hourly.windspeed_10m[i],
  rainChance: hourly.precipitation_probability[i],
  outfit: suggestOutfitEmojis(
    hourly.temperature_2m[i],
    hourly.windspeed_10m[i],
    getWeatherCondition(hourly.weathercode[i])
  ),
}));

setHourlyForecast(hourlyData);

// Reset animation values
hourAnims.length = 0;

hourlyData.forEach(() => {
  hourAnims.push(new Animated.Value(0));
});

// Tips
if (condition.includes('Rain')) {
  setTip('Don't forget your umbrella! ☔');
} else if (condition.includes('Snow')) {
  setTip('Bundle up, it's snowy! ❄');
} else if (temperature_2m_max[0] > 25) {
  setTip('Perfect day for shorts! 🍹🩴');
} else {
  setTip('Have a comfortable day! 😊');
}

// Animate stats
Animated.timing(statAnim, {
  toValue: 1,
  duration: 800,
  useNativeDriver: true,
}).start();

// Animate hourly cards

```

```

    Animated.stagger(
      100,
      hourAnims.map(anim =>
        Animated.timing(anim, {
          toValue: 1,
          duration: 500,
          useNativeDriver: true,
        })
      )
    ).start();

  } catch (err) {
    console.log('Weather Error:', err.response?.data || err.message);
    alert('Error fetching weather data');
  }

  setLoading(false);
};

const bgColor = todayWeather
  ? todayWeather.tempMax > 25 ? '#FFD700' : todayWeather.tempMax >= 15 ?
    '#87cefa' : '#4682b4'
  : '#87cefa';

const fallingEmoji = todayWeather
  ? ['Light Rain', 'Drizzle', 'Showers'].includes(todayWeather.condition)
  ? '💧'
  : ['Heavy Rain', 'Heavy
Showers', 'Thunderstorm'].includes(todayWeather.condition) ? '🌧️'
  : todayWeather.condition === 'Snow' ? '❄️' : null
  : null;

return (
  <ScrollView style={[styles.container, { backgroundColor: bgColor }]}
    contentContainerStyle={{ paddingBottom: 20 }}>
    { /* Header */ }
    <View style={styles.header}>
      <Text style={styles.title}>🌤️ Wear the Weather</Text>
      <TouchableOpacity onPress={() => setMenuVisible(true)}>
        <Text style={styles.menuText}>☰</Text>
      </TouchableOpacity>
    </View>
  </ScrollView>
);

```

```

    {/* City input */}
    <TextInput style={styles.input} placeholder="Enter city"
value={city} onChangeText={setCity} />

    {/* Button */}
    <View style={styles.buttonWrapper}>
      <TouchableOpacity style={styles.getOutfitButton}
onPress={getTodayWeather} disabled={loading}>
        <Text style={styles.getOutfitButtonText}>{loading ? 'Loading...'
: "Get Today's Outfit"}</Text>
      </TouchableOpacity>
    </View>

    {/* City Header */}
    {todayWeather && city && (
      <View style={styles.cityHeader}>
        <Text style={styles.cityName}>{city}
{getWeatherEmoji(todayWeather.condition)}</Text>
      </View>
    )}

    {/* Falling emojis */}
    {fallingEmoji && Array.from({ length: 10 }).map((_, i) =>
<FallingEmoji key={i} emoji={fallingEmoji} />)}

    {/* Today's Weather Header */}
    {todayWeather && <Text style={styles.todayHeader}>Today's
Weather</Text>}

    {/* Today's weather card */}
    {todayWeather && (
      <View style={styles.todayCard}>
        <Text style={styles.info}>🌡️ {todayWeather.tempMin}°C -
{todayWeather.tempMax}°C</Text>
        <Text style={styles.info}>☁️ {todayWeather.condition}
{getWeatherEmoji(todayWeather.condition)}</Text>
        <Text style={styles.outfit}>{todayWeather.outfit}</Text>
        <Text style={styles.tip}>{tip}</Text>
      </View>
    )}

```

```

    /* Dashboard Stats (animated) */
    {todayWeather && (
      <Animated.View style={ [styles.dashboardStats, {
        opacity: statAnim,
        transform: [{ translateY: statAnim.interpolate({
inputRange:[0,1], outputRange:[20,0] }) }]
      }]}>
        <View style={styles.statItem}>
          <Text style={styles.statEmoji}>🌤️</Text>
          <Text
style={styles.statText}>{todayWeather.rainChance}%</Text>
          <Text style={styles.statLabel}>Chance of Rain</Text>
        </View>
        <View style={styles.statItem}>
          <Text style={styles.statEmoji}>☀️</Text>
          <Text style={styles.statText}>{todayWeather.uv}</Text>
          <Text style={styles.statLabel}>UV Index</Text>
        </View>
        <View style={styles.statItem}>
          <Text style={styles.statEmoji}>🌀</Text>
          <Text style={styles.statText}>{todayWeather.wind} km/h</Text>
          <Text style={styles.statLabel}>Wind</Text>
        </View>
      </Animated.View>
    )}

    /* Hourly Forecast (animated cards) */
    {hourlyForecast.length > 0 && (
      <View style={{ marginTop: 20 }}>
        <Text style={styles.hourlyHeader}>Hourly Forecast</Text>
        <ScrollView horizontal showsHorizontalScrollIndicator={false}
style={{ marginTop: 10 }}>
          {hourlyForecast.map((hour, i) => (
            <Animated.View key={i} style={ [styles.hourCard, {
              opacity: hourAnims[i] || 1,
              transform: [{ scale: hourAnims[i] ?
hourAnims[i].interpolate({ inputRange:[0,1], outputRange:[0.8,1] }) : 1 }]
            }]}>
              <Text
style={styles.hourEmoji}>{getWeatherEmoji(getWeatherCondition(hour.code))}</Text>
              <Text style={styles.hourTemp}>{hour.temp}°C</Text>
              <Text style={styles.hourOutfit}>{hour.outfit}</Text>
            </Animated.View>
          ))}
        </ScrollView>
      </View>
    )}

```

```

        </Animated.View>
      )})
    </ScrollView>
  </View>
})

  { /* Dropdown menu */
    <Modal transparent visible={menuVisible} animationType="fade"
onRequestClose={() => setMenuVisible(false)}>
      <Pressable style={styles.modalOverlay} onPress={() =>
setMenuVisible(false)}>
        <View style={styles.dropdownMenu}>
          <TouchableOpacity style={styles.dropdownItem} onPress={() => {
setMenuVisible(false); navigation.navigate('Home'); }}>
            <Text style={styles.dropdownText}>Home</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.dropdownItem} onPress={() => {
setMenuVisible(false); navigation.navigate('Tips'); }}>
            <Text style={styles.dropdownText}>Tips</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.dropdownItem} onPress={() => {
setMenuVisible(false); navigation.navigate('Forecast'); }}>
            <Text style={styles.dropdownText}>Forecast</Text>
          </TouchableOpacity>
        </View>
      </Pressable>
    </Modal>
  </ScrollView>
);
}

const styles = StyleSheet.create({
  container: { flex:1, padding:20 },
  header: { flexDirection:'row', justifyContent:'space-between',
alignItems:'center', marginBottom:20 },
  title: { fontSize:28, fontWeight:'bold', color:'#fff' },
  menuText: { fontSize:28, color:'#fff' },
  input: { backgroundColor:'#fff', padding:10, borderRadius:8,
marginBottom:10 },
  buttonWrapper: { marginBottom: 10, borderRadius: 8, overflow: 'hidden'
},
  getOutfitButton: { backgroundColor:'rgba(255,255,255,0.85)',
paddingVertical:12, borderRadius:8, alignItems:'center' },

```

```

    getOutfitButtonText: { color:'#000', fontSize:16, fontWeight:'bold' },
    cityHeader: { alignItems:'center', marginTop:15, marginBottom:10,
paddingHorizontal:10, paddingVertical:5,
backgroundColor:'rgba(255,255,255,0.4)', borderRadius:10 },
    cityName: { fontSize:22, fontWeight:'bold', color:'#000',
textAlign:'center' },
    todayHeader: { fontSize:22, fontWeight:'bold', color:'#fff',
marginTop:10, marginBottom:10, textAlign:'center' },
    todayCard: { marginTop:10, padding:20, borderRadius:12,
alignItems:'center', backgroundColor:'rgba(255,255,255,0.8)' },
    info: { fontSize:18, color:'#000', marginVertical:4 },
    outfit: { fontSize:20, color:'#000', marginTop:10, textAlign:'center' },
    tip: { fontSize:16, color:'#555', marginTop:10, fontStyle:'italic' },
    dashboardStats: { flexDirection:'row', justifyContent:'space-around',
marginTop:20, padding:15, backgroundColor:'rgba(255,255,255,0.8)',
borderRadius:12 },
    statItem: { alignItems:'center', flex:1 },
    statEmoji: { fontSize:28 },
    statText: { fontSize:18, fontWeight:'bold', color:'#000', marginTop:5 },
    statLabel: { fontSize:12, color:'#333', marginTop:2, textAlign:'center'
},
    hourlyHeader: { fontSize:18, fontWeight:'bold', color:'#fff' },
    hourCard: { backgroundColor:'rgba(255,255,255,0.8)', padding:10,
marginRight:10, borderRadius:10, alignItems:'center', width:100 },
    hourEmoji: { fontSize:24 },
    hourTemp: { fontSize:16, fontWeight:'bold', marginTop:5 },
    hourOutfit: { fontSize:12, marginTop:5, textAlign:'center' },
    modalOverlay: { flex:1, backgroundColor:'rgba(0,0,0,0.3)',
justifyContent:'flex-start', alignItems:'flex-end' },
    dropdownMenu: { backgroundColor:'#4682b4', marginTop:60, marginRight:20,
borderRadius:8, width:150 },
    dropdownItem: { padding:15, borderBottomWidth:1,
borderBottomColor:'#fff' },
    dropdownText: { color:'#fff', fontSize:16 },
  });

```

CM ASSIGN2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

81 /100

Marks

awarded

Total 81

1. Engagement with Literature Skills 10

- + Good use of citations included but limited academic evaluation or critical comparison. 8
- + Well-formatted reference list.

2. Knowledge and Understanding Skills 5, 10

T2 Q1: 5

- + Tech Start-up name, Logo, Business type (Legal classification) and a Business Strategy are documented. And Funding options considered.

T2 Q2:

+ **PEST analysis** summary includes some relevant considerations for each category and documents some key professional legal & ethical considerations but analysis remains largely explanatory and could provide stronger critical evaluation of business risks and ethical challenges. 8

+ Legal & ethical considerations are included but not each category is supported by relevant citations.

3. Cognitive and Intellectual Skills 5, 5, 5, 5

T3 Q1: - 3

Data collection method for practical feedback designed, and some feedback acquired and considered. Feedback discussion is brief and provides limited analysis of participant responses or resulting development changes

T3 Q2: + Project

management reflection (including review of weekly progress sheets) documented. Reflective evaluation of planning effectiveness and lessons learned could be explored in greater depth.

T3 Q3: - Skill

development reflection documented. Reflection is brief and focuses mainly on skills gained rather than evaluating development or future professional application.

T3 Q3: - Project's positive impact on society 4

is clearly described. Discussion is brief and wider social, accessibility or wellbeing impacts are not explored in depth.

3

3

4. Practical Application Skills 10, 30, 10

T1 Q1: + 8

Data collection method for initial feedback designed, and some feedback acquired and considered. Stronger evaluation of results and deeper discussion of findings would strengthen analysis.

+ Data collection tool and question set are documented.

T1 Q2: + key features of the **implementation** process and practical output have been clearly documented. Some implementation discussion remains descriptive and greater explanation of system architecture or AI logic would strengthen technical depth further.
– Some additional practical implementation work could be included to gain the additional marks in this category.

Progress sheets (Appendix): +
Documentation of weekly progress sheets.

5.
Transferable Skills for Life and
Professional Practice 5

· + Very good report presentation structure 4
and format

·

+

Title page & Table of contents.

·

+

Introduction & Conclusion.

·

+

Tasks complete.

Appendix: CPD course.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31
