Sprint	User Story/Technical Story	Acceptance Criteria
		Users can input their location manually.
		Users can choose from a list of predefined cities.
		The system retrieves and displays current temperature.
		The system retrieves and displays current humidity.
		The system retrieves and displays current weather condition.
	User Story 1: As a user, I want to input my current location to get real-time weather data.	Handle invalid locations or errors gracefully.
		Users have the option to allow location-based weather data.
		The system requests geolocation permission.
		The system retrieves weather data automatically.
	User Story 2: As a user, I want the system to detect my location automatically so I can quickly get weather info.	Handle errors when location permission is denied.
		The homepage displays the temperature clearly.
		The homepage displays the humidity clearly.
		The homepage displays the wind speed clearly.
		The homepage displays the weather condition clearly.
Sprint 1	User Story 3: As a user, I want to see basic weather details clearly on the homepage.	Ensure a simple and user-friendly layout.
		Users can refresh the weather data via a refresh button.
	User Story 4: As a user, I want to refresh weather data manually.	The page updates with the latest weather information.
		Ensure the website layout is responsive on desktop.
		Ensure the website layout is responsive on mobile devices.
	User Story 5: As a user, I want the website to be responsive for mobile devices.	Weather data should be clearly visible on all devices.
		Create a simple homepage with location input fields.
		Create buttons for location input and refresh.
	Technical Story 1: Set up basic UI/UX design for weather data display.	Display weather info in an organized, easy-to-read format.
		Successfully fetch weather data from an external API.
		Ensure API integration is stable.
Sprint 1	Technical Story 2: Integrate a weather API to fetch real-time weather data.	Handle errors for invalid locations.
Sprint 2	User Story 6: As a user, I want to receive an outfit suggestion based on the current weather conditions.	Display a recommended outfit for cold weather (e.g., coat).
		Display a recommended outfit for warm weather (e.g., light clothes).
		Adapt outfit suggestions based on rain.
		Adapt outfit suggestions based on snow.
		Adapt outfit suggestions based on wind.
Sprint 2	User Story 7: As a user, I want to plan an outfit for a future date or journey.	Users can input a future date for weather forecast.
		Display outfit suggestions based on future forecast.
		Allow users to select time of day for more accurate planning.
Sprint 2	User Story 8: As a user, I want to customize my outfit preferences based on my comfort level.	Users can set personal preferences for warmer outfits.
		Users can set personal preferences for cooler outfits.
		Tailor outfit recommendations based on preferences.
Sprint 2	User Story 9: As a user, I want to receive clothing layer suggestions based on cold weather.	Provide layered clothing suggestions for cold weather.
		Adjust layer recommendations based on rain.
		Adjust layer recommendations based on cold temperatures.
Sprint 2	User Story 10: As a user, I want to check the weather and outfit recommendations for multiple days during my trip.	Users can plan outfits for multiple days.
		Retrieve weather forecasts for multiple dates.
		Provide outfit suggestions based on forecasts for selected dates.
Sprint 2	Technical Story 3: Implement a weather forecast-based recommendation system.	Integrate weather forecast data for future dates.
		Provide accurate outfit suggestions based on forecasted weather.
Sprint 2	Technical Story 4: Build a UI for future date outfit planning and personal outfit customization.	Add a user-friendly interface for selecting future dates.
		Add a user-friendly interface for customizing outfit preferences.
		Ensure seamless outfit planning functionality.
Sprint 3	User Story 11: As a user, I want to create an account and log in so that I can save my preferences and outfit history.	Users can create an account with basic details (email/password).
		Users can log in and out.
		User data is saved (outfit preferences, history).
		User data is retrieved when logged in.
Sprint 3	User Story 12: As a user, I want to view previous outfit suggestions for future reference.	Users can view past outfit suggestions based on previous weather conditions.
		Outfit suggestions for specific dates/locations can be saved.
Sprint 3	User Story 13: As a user, I want to receive tailored outfit suggestions based on past choices and preferences.	Provide personalized outfit suggestions based on past user selections.
Sprint 3	Technical Story 5: Implement a simple user authentication system.	Tailor recommendations using past weather and outfit data.
		Set up secure user registration system.
		Set up secure login system.
		Store user data (preferences, outfit history).
Sprint 3	Technical Story 6: Finalize UI/UX improvements and make the website fully responsive.	Enhance UI/UX for a polished experience.
		Ensure website works smoothly across different devices.
		Ensure full responsiveness across all screen sizes.