WEATHER WEAR

Presented By Team - 3

Bad Ideas

AGENDA

1. Team Members Roles and Responsibilities
2. Improvements made from Professor Feedback
3. Project Description
4. Team Working Agreement
5. Personas
6. MVP
7. Technologies
8. Algorithms
9. Diagrams
10. Product backlog
11. Test Cases
12. Stories Completed
13. Stories not completed
14. Metrics
15. Retrospective
16. Sprint 2
17. Project Demo
18. Github Link
19. Live Application Demo

THIS IS OUR TEAM

Helloween James

Project Manager & Machine Learning Engineer

Gracia Betty Jebaraj Daniel

Frontend Developer & UI/UX Designer

Tejaswini Kandyala

Frontend Developer

THIS IS OUR TEAM

Bharadwaj Reddy Asireddy

Backend Developer & UI/UX Designer

Mohith Durga Srinivas Tripuramallu

Backend Developer

Sai Rajeswari Ghanta

Quality Assurance Tester

Improvements made from professor feedback

Tools Alignment:

Adjusted the alignment of tools on the wikipage for improved readability and user experience, based on feedback.

Sprint planning:

Transitioned from a 2-day project schedule in the last sprint to a detailed sprint plan, visualized for better tracking and transparency.

PROJECT DESCRIPTION

Project Name:	**Weather Wear**
Team:	Bad Ideas
Project Description:	Predicts weather and suggests clothing **For** Users **who** do not know have the time to wear clothes or buy them according to the weather **the** Weather Wear Application **is a** Weather-Based Clothing Recommendation System **that** it suggests clothing options based on current weather forecasts using advanced machine learning. **unlike** other weather applications or clothing accessories **our application** helps users pick the best outfits for their travel plans, activities, or packed schedules. Considering the weather, it makes getting dressed easier whether you're on the go or planning a trip.
Benefit Outcomes:	• Saves time and allows users on better planning. • Users need not be worried about them being overdressed or underdressed. • This makes them prepared for the unpredictable weather. • Makes the selection of clothes seamless
Github Link:	https://github.com/htmw/2024F-Bad-Ideas/wiki

TEAM AGREEMENT

1. Team Information

- **Project Title:** Weather Wear
- **Team Name:** Bad Ideas
- **Team Members and Roles :**
  - Helloween James – Project Manager & Machine learning engineer
  - Gracia Betty Jebaraj Daniel – Front-end developer & UI/UX Designer
  - Tejaswini Kandyala – Front-end developer
  - Mohith Durga Srinivas Tripuramallu – Back-end Developer
  - Bharadwaj Reddy Asireddy - Back-end Developer & UI/UX Designer
  - Sai Rajeswari Ghanta - Quality Assurance (QA) Tester

2. Meetings and Communication

We will meet twice a week and we shall use what's app, email, outlook and Zoom as our communication for any updates and response should be within 2-3hrs.

3. Work Distribution

Everyone in the team agrees to share the work equally and if any member fells overwhelmed, we will redistribute the tasks.

TEAM AGREEMENT

4. Conflict Resolution

· If we have any sort of disagreement in the tasks, we'll talk it through vote.
· If the conflict cannot be resolved internally, we will reach professor for an advice

5. Deadlines

· We will set deadlines for each task and everyone agrees to stick to them and if someone can't meet a deadline, they should inform the team in advance.

· We agree to submit everything on time, with everyone's contribution.

6. Signatures

- Gracia Betty Jebaraj Daniel
- Tejaswini Kandyala
- Helloween James
- Mohith Durga Srinivas Tripuramallu
- Bharadwaj Reddy Asireddy
- Sai Rajeswari Ghanta

PERSONAS

HANNAH

This is Hannah, Age 28. She stays in New York City, NY and she is Product Manager.

- **Life Style:** She commutes everydays and she meets a lot of clients.
- **Weather and Clothing:** NYC weather is totally unpredictable as it can rain, be sunny or be cold anytime. She likes wearing formal clothes.
- **Technology usage:** She loves trying out new apps.
- She is busy person and always wears uncomfortable dress which does not align well with weather, so she needs something which can set her to wear clothes which doesn't waste her time.

JASON

This is Jason, Age 34. He stays in Seattle, WA, and he is a Software Engineer.

- **Life Style:** He commutes to the office twice a week and enjoys biking when the weather permits.
- **Weather and Clothing:** Seattle's weather is unpredictable, with frequent rain and sudden changes. He prefers casual and comfortable clothing but often finds himself either underdressed for cold weather or overdressed when it warms up unexpectedly.
- **Technology usage:** He is tech-savvy and enjoys trying out new apps to improve efficiency.
- He frequently faces issue with choosing the right clothes due to the constant change in weather and he needs a solution that helps him dress appropriately for the day.

ANANYA

This is Ananya, Age 22. She stays in Mumbai, India, and she is a College Student.

- **Life Style:** She walks around campus a lot, balancing her classes, internships, and extracurricular activities.
- **Weather and Clothing:** Mumbai's weather can be unpredictable, with sudden rain showers or humid heat. She likes trendy, comfortable clothes but struggles to dress appropriately when the weather changes quickly.
- **Technology usage:** She uses multiple apps to manage her busy schedule and lifestyle.
- She often finds herself either drenched in rain or overheated due to sudden weather changes and needs something that will help her choose the right outfit for the day without wasting time.

Minimum Viable Product

- **Location-Based Weather Data**: Fetch and display real-time weather.
- **Outfit Suggestions**: Recommend outfits based on weather.
- **Future Outfit Planning**: Plan outfits for future dates.
- **User Preferences**: Customize outfit recommendations.
- **Responsive Design**: Optimized for mobile and desktop.

TECHNOLOGIES

Frontend

- Next JS
- Shadcn
- Tailwind CSS

Backend

- Flask
- Rapid API

Database

- MongoDB

TECHNOLOGIES

Deployment

- Vercel
- AWS

Machine Learning

- Scikit Learn
- Pytorch

Tools

- Figma
- VSCode
- Codespaces
- Github
- Postman

ABOUT FRONTEND TECHNOLOGIES

So here next js is used for building the website because next js is one of the best frontend frameworks and with integration with shadcn and tailwind css, building minimalistic website easier.

ABOUT BACKEND TECHNOLOGIES

RapidAPI offers a variety of weather APIs that provide all the data you could need. These APIs are great for delivering accurate and up-to-date weather information.

For our project, we chose one of these weather APIs from RapidAPI to boost our data capabilities.

ABOUT DATABASE TECHNOLOGIES

MongoDB is one the best NoSQL database and as the application data can also be unstructured, this would be best option to choose it. This also have fast retrieval (read operations) compared to SQL databases, so this is better choice for this application.

ABOUT DEPLOYMENT TECHNOLOGIES

Here, first vercel is used for deploying the frontend part of the application and aws is used for deploying backend and machine learning as well because NextJS works seamless in vercel, as vercel are the one who developed Next js.

ABOUT MACHINE LEARNING TECHNOLOGIES

So here scikit learning is primarily used for building the basic version of the machine learning algorithms and in later on stages, Pytorch is used for build even more accurate version of the machine learning which understands the weather even more better and predicts it well.

ABOUT TOOLS TECHNOLOGIES

Figma is mainly used for prototyping and design how the product looks like. VSCode is where actually the code is written. Codespaces is where the collaboration code is done, for pair programming. Github is where whole code is hosted. Postman is for testing the API endpoints.

ALGORITHMS

Predicting Weather

Random forest regressor

- Used to predict weather conditions like temperature or humidity by learning patterns from historical weather data through decision trees that capture non-linear relationships.

LSTM

- A recurrent neural network used to predict future weather by capturing long-term dependencies and patterns in sequential weather data over time.

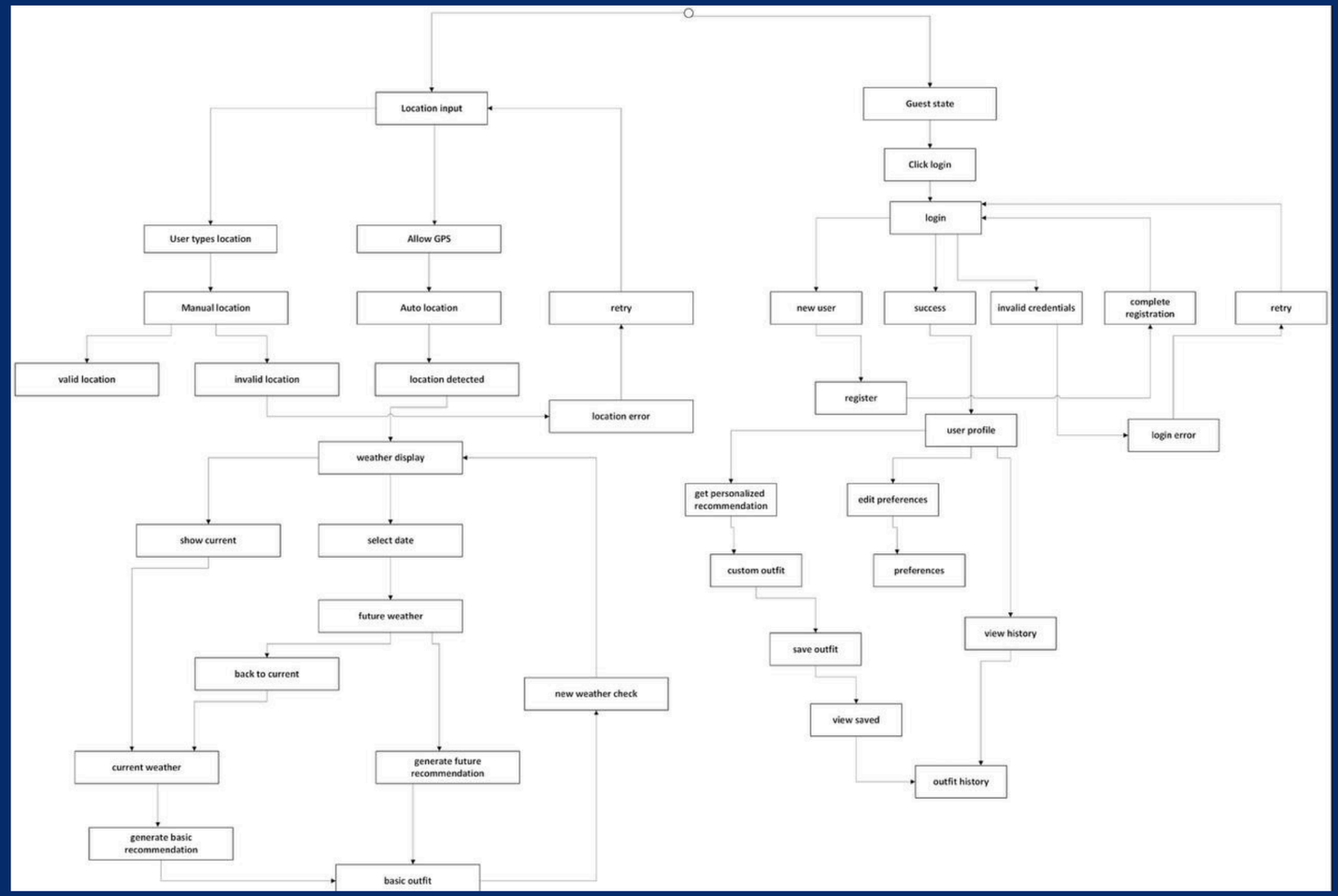
Recommendation

KNN (k-nearest neighbours)

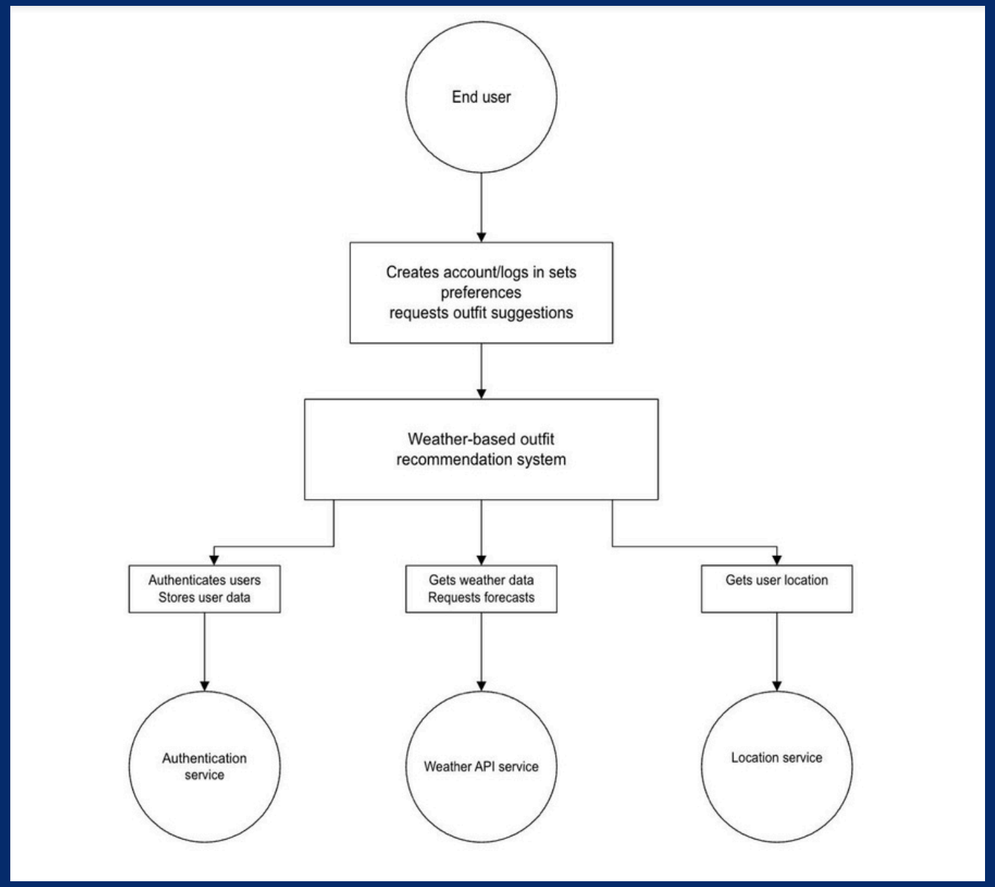
- Used to recommend outfits by finding similar users (or outfits) based on user preferences and weather conditions, suggesting clothing choices that worked for users with similar profiles or conditions.

DIAGRAMS

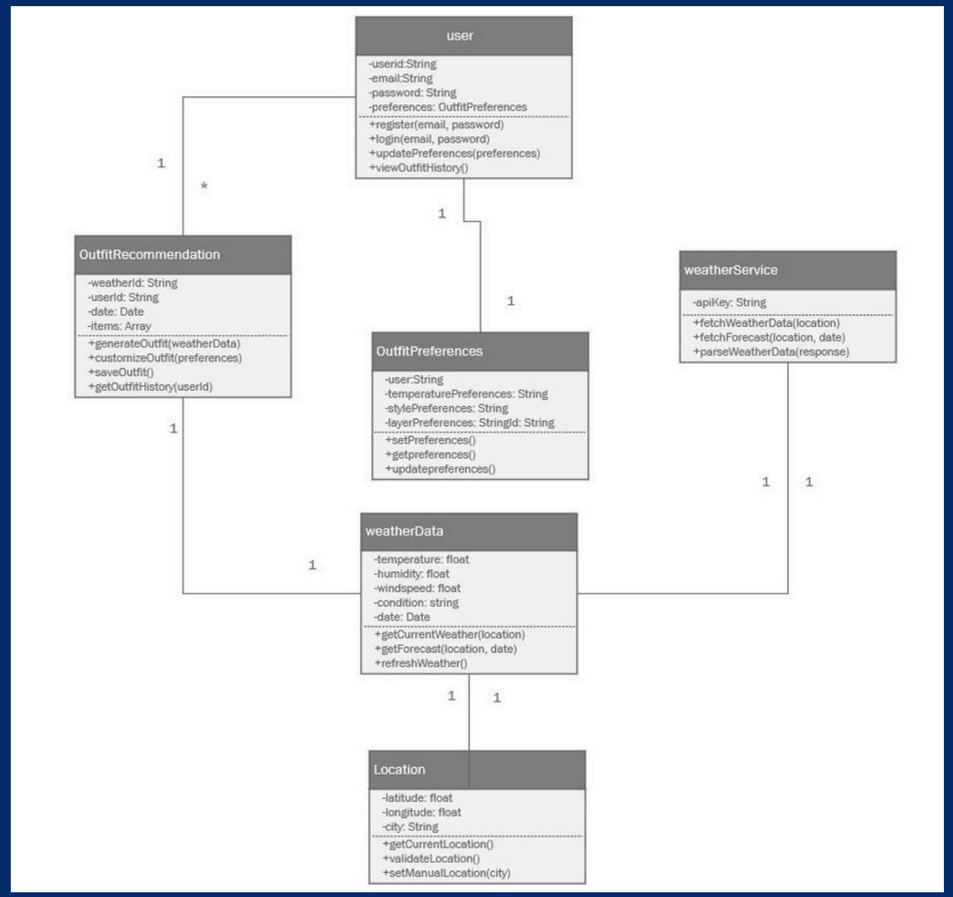
State Diagram



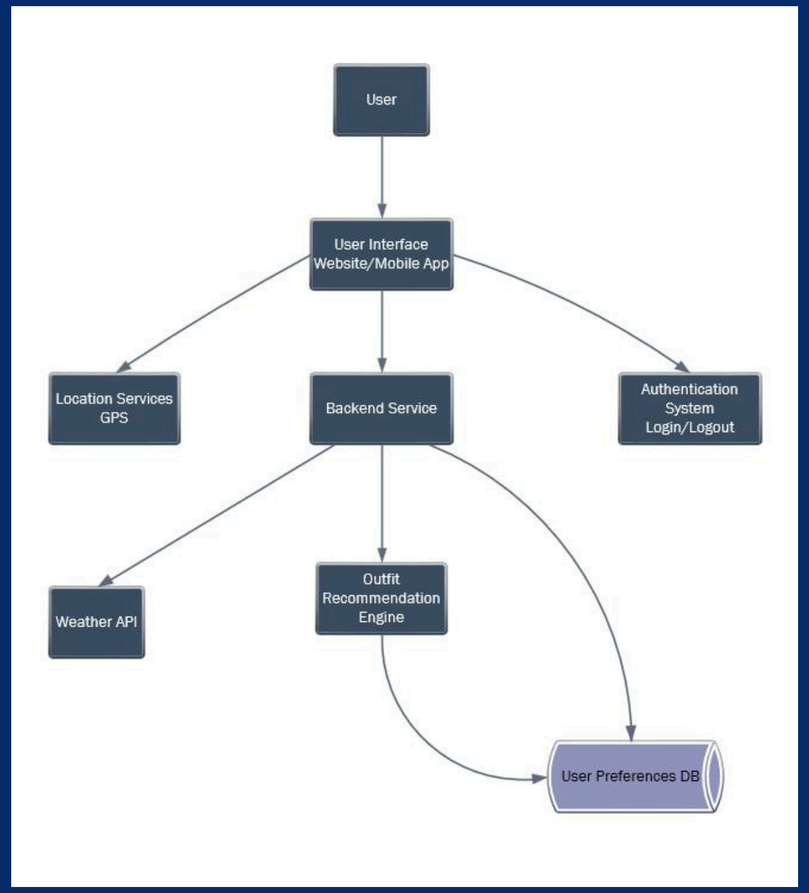
Context diagram



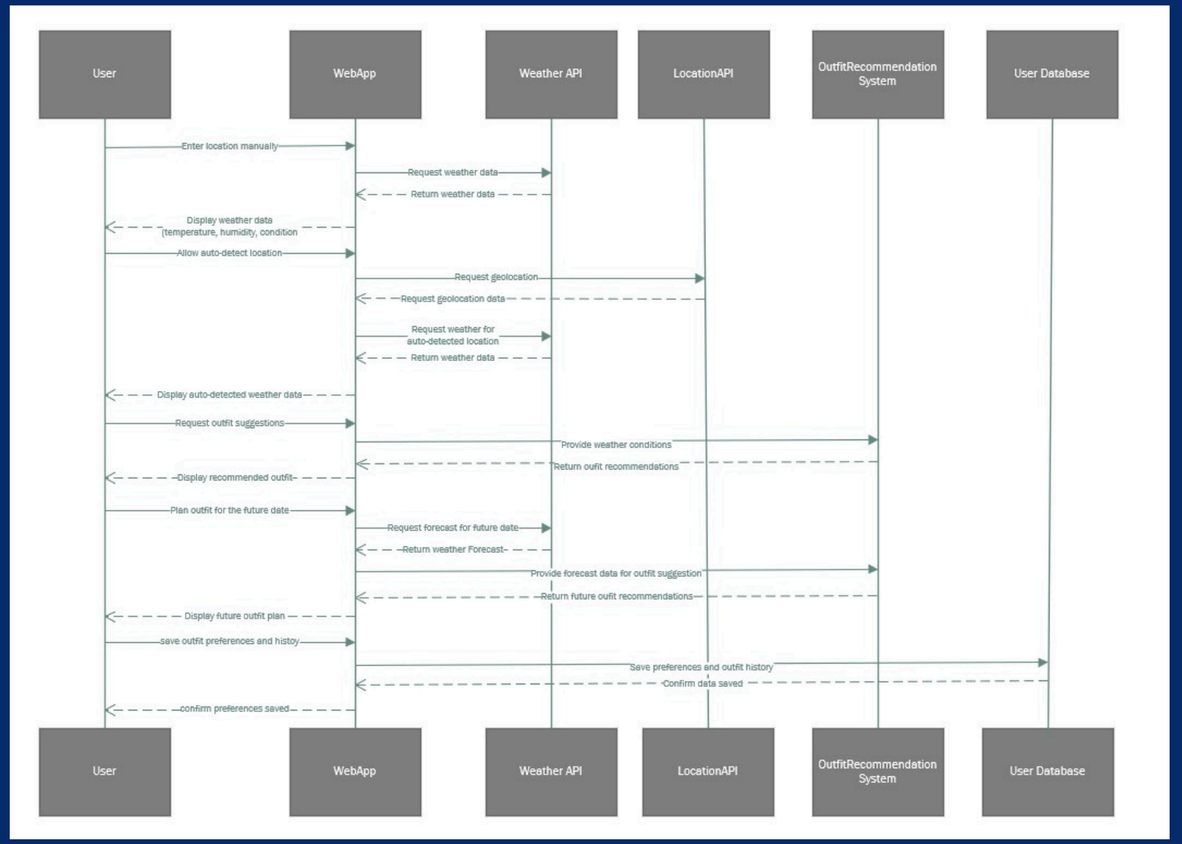
Class diagram



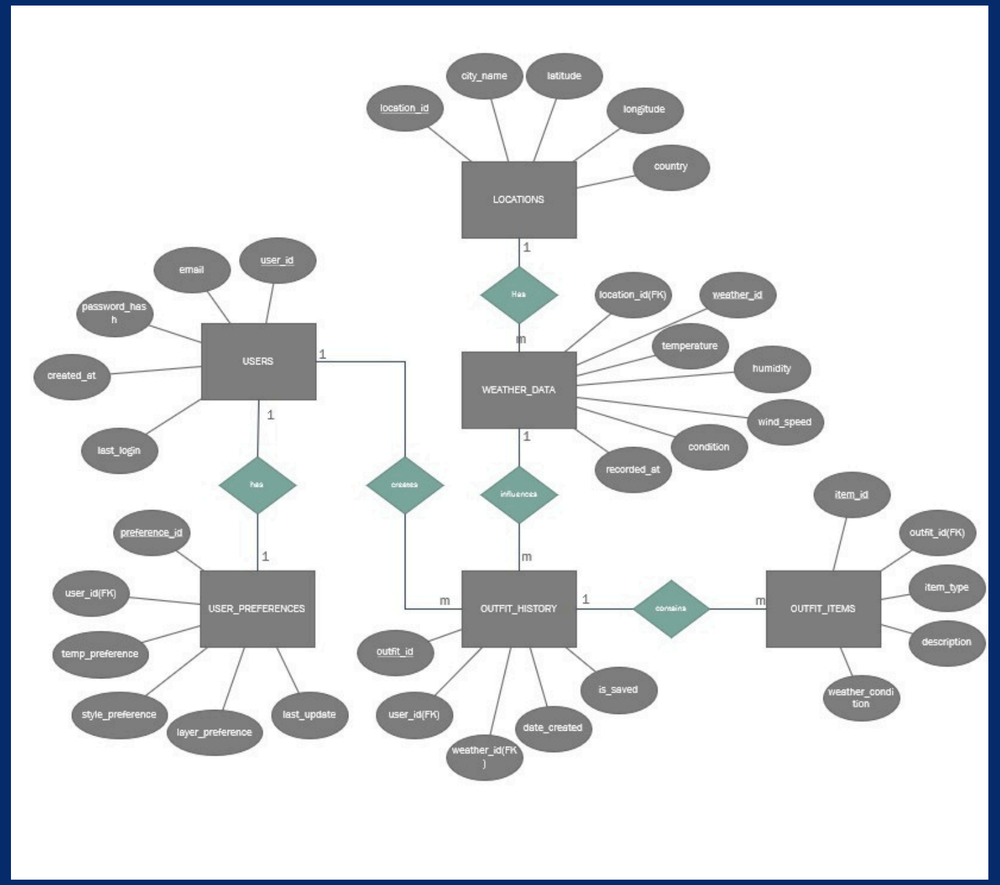
Architecture diagram



Sequence Diagram



ER Diagram


city_name
latitude
location_id
longitude
country
LOCATIONS
1
Has
m
email
user_id
password_hash
location_id(FK)
weather_id
temperature
USERS
1
WEATHER_DATA
humidity
created_at
wind_speed
1
1
last_login
condition
recorded_at
has
creates
influences
item_id
outfit_id(FK)
preference_id
1
m
item_type
user_id(FK)
USER_PREFERENCES
m
OUTFIT_HISTORY
1
contains
m
OUTFIT_ITEMS
description
outfit_id
temp_preference
is_saved
weather_condition
user_id(FK)
style_preference
last_update
layer_preference
date_created
weather_id(FK)

Product Backlog

Sprint	***User Story/Technical Story***	***Acceptance Criteria***
Sprint 1	User Story 1: As a user, I want to input my current location to get real-time weather data.	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.
		Handle invalid locations or errors gracefully.
	User Story 2: As a user, I want the system to detect my location automatically so I can quickly get weather info.	Users have the option to allow location-based weather data.
		The system requests geolocation permission.
		The system retrieves weather data automatically.
		Handle errors when location permission is denied.
	User Story 3: As a user, I want to see basic weather details clearly on the homepage.	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.
		Ensure a simple and user-friendly layout.

Product Backlog

Sprint	***User Story/Technical Story***	**Acceptance Criteria**
Sprint 1	User Story 4: As a user, I want to refresh weather data manually.	Users can refresh the weather data via a refresh button.
		The page updates with the latest weather information.
	User Story 5: As a user, I want the website to be responsive for mobile devices.	Ensure the website layout is responsive on desktop.
		Ensure the website layout is responsive on mobile devices.
		Weather data should be clearly visible on all devices.
	Technical Story 1: Set up basic UI/UX design for weather data display.	Create a simple homepage with location input fields.
		Create buttons for location input and refresh.
		Display weather info in an organized, easy-to-read format.
	Technical Story 2: Integrate a weather API to fetch real-time weather data.	Successfully fetch weather data from an external API.
		Ensure API integration is stable.
		Handle errors for invalid locations.

Product Backlog

Sprint	***User Story/Technical Story***	***Acceptance Criteria***
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.
	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.
	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.

Product Backlog

Sprint	***User Story/Technical Story***	***Acceptance Criteria***
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.
	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.
	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.
	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.

Product Backlog

Sprint	***User Story/Technical Story***	***Acceptance Criteria***
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.
	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.
	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.
		Tailor recommendations using past weather and outfit data.
	Technical Story 5: Implement a simple user authentication system.	Set up secure user registration system.
		Set up secure login system.
		Store user data (preferences, outfit history).

Product Backlog

Sprint	***User Story/Technical Story***	***Acceptance Criteria***
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.

Sprint 1

Story ID	***User Story/Task Description***	***Acceptance Criteria***	***Story Points***
US1	User Story 1: Input location to get real-time weather data	Users can input location manually.	5
		Choose from predefined cities.	
		Display temperature.	
		Display humidity.	
		Display weather condition.	
		Handle errors for invalid locations.	
US2	User Story 2: Detect user location automatically	Request geolocation permission.	3
		Retrieve weather data automatically.	
		Handle errors if permission is denied.	
US3	User Story 3: Display basic weather details clearly on the homepage	Show temperature.	3
		Show humidity.	
		Show wind speed.	
		Show weather condition.	
		Ensure user-friendly layout.	

Sprint 1

Story ID	***User Story/Task Description***	***Acceptance Criteria***	***Story Points***
US4	User Story 4: Manually refresh weather data	Users can refresh data via a button.	**2**
		Latest weather is displayed.	
US5	User Story 5: Make website responsive for mobile devices	Website works on desktop.	**5**
		Website works on mobile devices.	
		Weather data clearly visible on all devices.	
TS1	Technical Story 1: Set up basic UI/UX design for weather data display	Create homepage with input fields.	**3**
		Add buttons for location input and refresh.	
		Display weather data neatly.	
TS2	Technical Story 2: Integrate a weather API to fetch real-time data	Successfully fetch real-time weather data from an external API.	**5**
		Handle errors for invalid locations.	

Test Cases - 1

Story ID	***Test Case***	***Expected***	***Tested***	***Pass/Fail***
US1	Verify manual location input.	Manual location input is accepted.	Tested on desktop	Pass
	Test with valid and invalid locations.	Valid locations show weather; invalid locations show error message.	Tested on mobile	Pass
	Confirm weather data display.	Correct weather data (temperature, humidity, condition) is displayed.	Tested on both	Pass
US2	Test geolocation permission request.	System requests user location permission.	Tested on desktop	Pass
	Validate automatic weather retrieval.	Weather is automatically fetched based on user location.	Tested on mobile	Pass
	Check error handling.	Error message displays if geolocation permission is denied.	Tested on desktop	Pass
US3	Confirm that weather details (temperature, humidity, wind, condition) are shown correctly.	All weather details are displayed correctly on the homepage.	Tested on both	Pass

Test Cases - 1

Story ID	***Test Case***	***Expected***	***Tested***	***Pass/Fail***
US4	Verify that the refresh button fetches updated weather data.	Weather data updates after refresh button is clicked.	Tested on desktop	Pass
	Ensure page refreshes correctly.	The page refreshes and shows the latest data.	Tested on mobile	Pass
US5	Test responsiveness on various devices (mobile, tablet, desktop).	Website is responsive and adapts to different screen sizes.	Tested on tablet	Pass
	Check if layout is clear on all devices.	Layout remains clear and usable across all devices.	Tested on mobile	Pass
TS1	Test UI elements (input fields, buttons) on the homepage.	Input fields and buttons work as expected.	Tested on both	Pass
	Ensure layout is intuitive and easy to navigate.	Layout is easy to use and navigate for users.	Tested on desktop	Pass
TS2	Validate API integration.	Weather data is successfully retrieved from the external API.	Tested on both	Pass
	Test weather retrieval for valid and invalid locations.	Valid locations show data, invalid ones trigger error handling.	Tested on desktop	Pass
	Handle API errors.	System gracefully handles API errors (e.g., network failure).	Tested on both	Pass

Stories Completed

Story ID	***User Story/Task Description***	***Story Points***
US1	Input location to get real-time weather data	5
US2	Detect user location automatically	3
US3	Display basic weather details clearly on the homepage	3
US4	Manually refresh weather data	2
US5	Make website responsive for mobile devices	5
TS1	Set up basic UI/UX design for weather data display	3
TS2	Integrate a weather API to fetch real-time weather data	5

Metrics

Team Velocity (This Sprint)

- **Total Story Points Committed**: 26 points
- **Total Story Points Completed**: 26 points

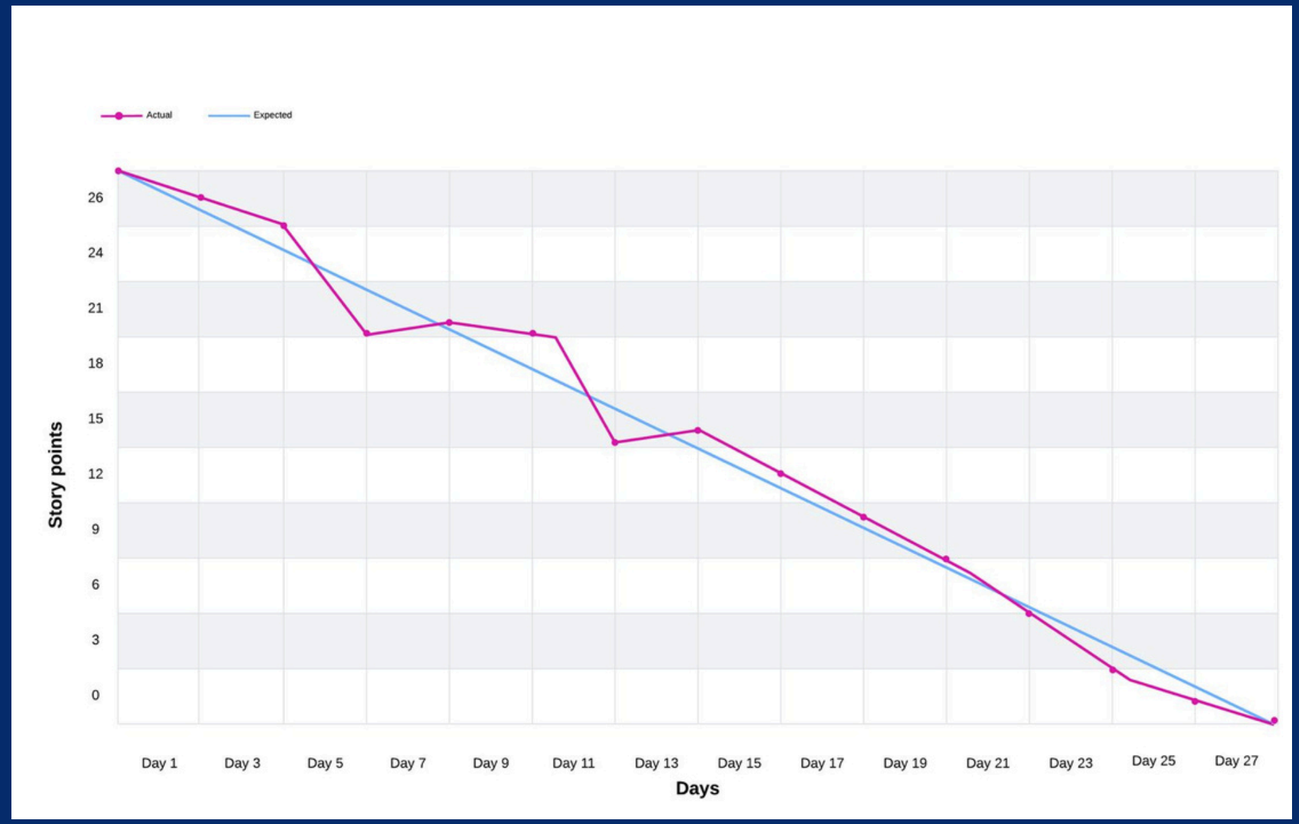
Committed to Completed

- **Story Points Committed: 26**
- **Story Points Completed: 26**

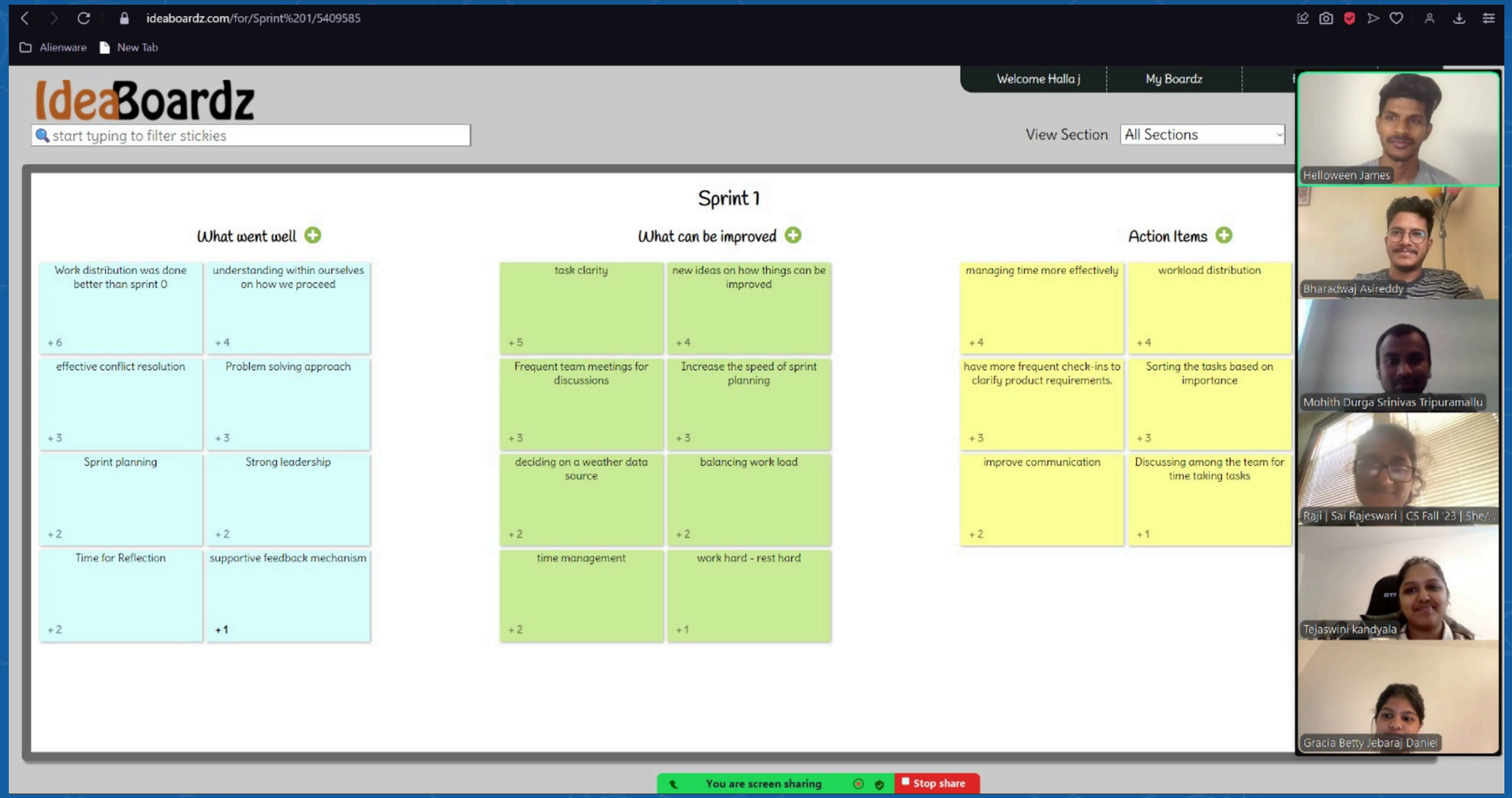
The ratio is:
26 / 26 = 100%

Metrics

Burn Down Chart



RETROSPECTIVE



SPRINT PLANNING

Sprint 0

Task	Status	Due date	Priority	Time Projection
Discovering new Project Ideas	Done	~~Sep 4~~	Medium	5 Days
Assigning Team Roles	Done	~~Sep 7~~	Medium	1 Day
Teem Agreement and Weekly Meetings	Done	~~Sep 8~~	High	1 Day
Project idea & Algorithms	Done	~~Sep 15~~	Critical ⚠	6 Days
Github repository and Wiki-page Setup	Done	~~Sep 17~~	Medium	3 Days
Retrospective	Done	~~Sep 19~~	High	1 Day
Presentation	Done	~~Sep 21~~	High	2 Days
Sprint 1 Discussion	Done	~~Sep 24~~	Low	1 Day
+ Add task				
		Sep 4 - 24		

Bad Ideas - Weather Wear

Integrate

Main Table | Kanban | +

New task | Search | Person | Filter | Sort | Hide | Group by | ...

Sprint - 1

Task	Status	Due date	Priority	Time Projection
Main meeting - Discussion ...	Done	~~Sep 25~~	Low	5 Days
Assigning work & balancing...	Working on it	Sep 28	High	3 Day
Research on Existing Models	Stuck	Oct 3	Medium	5 Days
User Stories	Working on it	Oct 5	Medium	2 Days
Commits to Github	Not Started	Oct 11	High	2 Days
Project Execution (MVP)	Not Started	Oct 17	Critical ⚠	5 Days
Retrospective	Not Started	Oct 21	High	1 Day
Presentation	Not Started	Oct 21	High	1 Day
+ Add task				
		Sep 25 - Oct 21		

Bad Ideas - Weather Wear

Integrate

Main Table | Kanban | +

New task | Search | Person | Filter | Sort | Hide | Group by | ...

Sprint - 1 8 Tasks

Task	Status	Due date	Priority	Time Projection
Main meeting - Discussion ...	Done	~~Sep 25~~	Low	5 Days
Assigning work & balancing...	Done	~~Sep 28~~	High	3 Day
Research on Existing Models	Done	~~Oct 3~~	Medium	5 Days
User Stories	Done	~~Oct 5~~	Medium	2 Days
Commits to Github	Done	~~Oct 11~~	High	2 Days
Project Execution (MVP)	Done	~~Oct 17~~	Critical ⚠	5 Days
Retrospective	Done	~~Oct 21~~	High	1 Day
Presentation	Done	~~Oct 21~~	High	1 Day
+ Add task				
		Sep 25 - Oct 21		

Sprint 2

Sprint 2

Task	Status	Due date	Priority	Time Projection
Technical Paper Discussion	Not Started	Oct 23	Medium	2 Days
Draft Technical Paper	Not Started	Oct 27	High	5 Days
Train ML Model & Refine	Not Started	Oct 30	High	6 Days
Add Planned Features & Changes	Not Started	Nov 3	Medium	6 days
Discuss and work on improvements	Not Started	Nov 7	Medium	4 Days
Retrospective	Not Started	Nov 16	High	1 Day
Presentation	Not Started	Nov 17	High	2 Days
Sprint 3 Discussion	Not Started	Nov 19	Low	1 Day
+ Add task				
		Oct 23 - Nov ...		

Sprint 3

Task	Status	Due date	Priority	Time Projection
Complete Application Build	Not Started	Nov 26	Critical	5 Days
Complete Project Demo	Not Started	Nov 27	Critical	5 Days
Complete Technical Paper	Not Started	Dec 1	Critical	5 Days
Retrospective	Not Started	Dec 4	Critical	2 Days
Project Presentation	Not Started	Dec 9	Critical	1 Day
+ Add task				
		Nov 26 - Dec 9		

Sprint 2

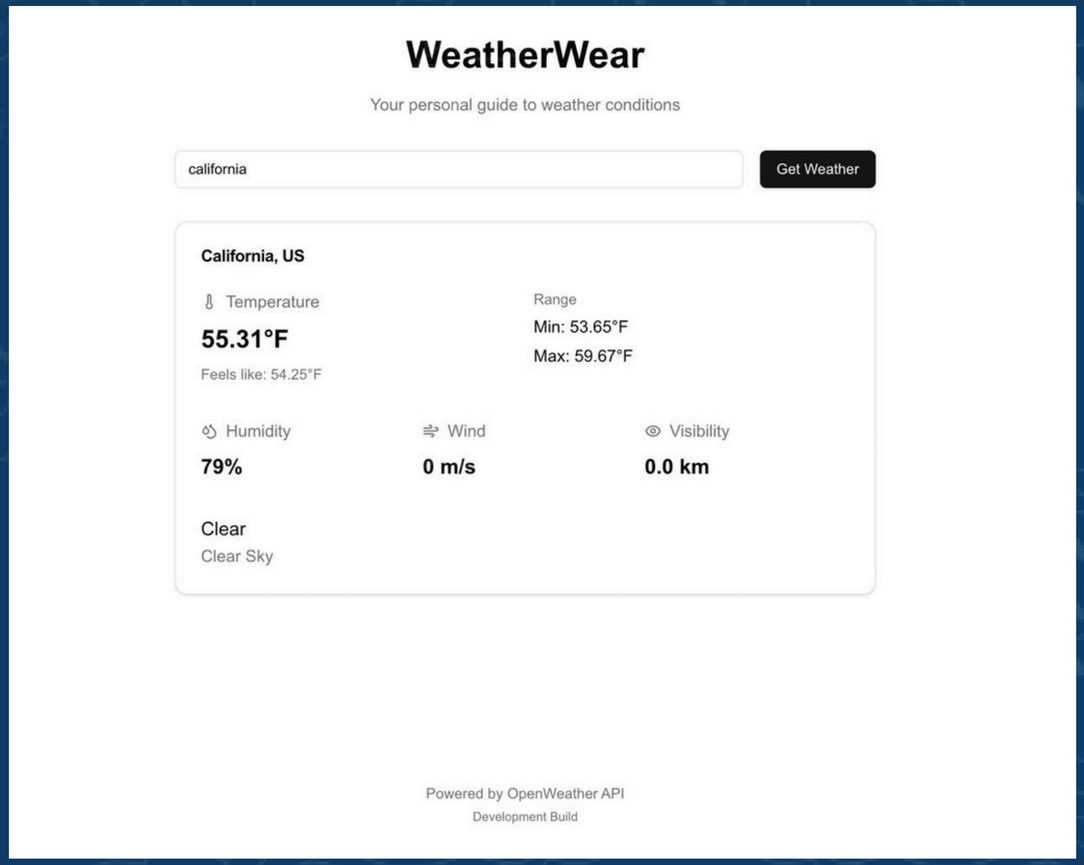
Story ID	***User Story/Task Description***	***Acceptance Criteria***	***Story Points***
US6	User Story 6: Receive outfit suggestions based on the current weather	Recommend outfit for cold weather.	**8**
		Recommend outfit for warm weather.	
		Adapt suggestions for rain.	
		Adapt suggestions for snow.	
		Adapt suggestions for wind.	
US7	User Story 7: Plan an outfit for a future date	Input future date.	**5**
		Display weather forecast.	
		Provide outfit suggestions based on forecast.	
US8	User Story 8: Customize outfit preferences based on comfort level	Set preferences for warmer outfits.	**3**
		Set preferences for cooler outfits.	
		Tailor recommendations based on preferences.	

Sprint 2

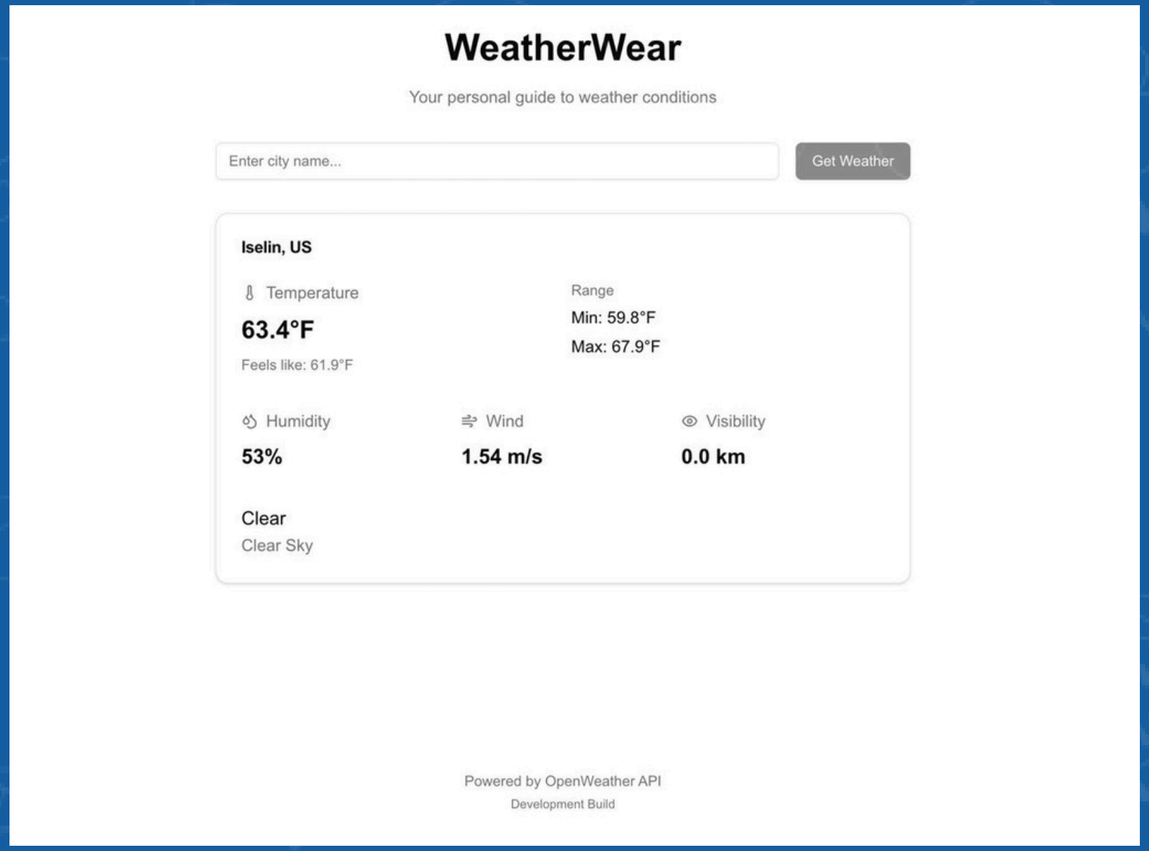
Story ID	***User Story/Task Description***	***Acceptance Criteria***	***Story Points***
US9	User Story 9: Receive clothing layer suggestions	Recommend layers for cold weather.	**5**
		Suggest rain gear.	
		Suggest cold weather accessories.	
US10	User Story 10: Check weather and outfit recommendations for multiple days during a trip	Input multiple future dates.	**8**
		Retrieve forecasts for multiple days.	
		Provide outfit suggestions for each date.	
TS3	Technical Story 3: Implement a weather forecast-based recommendation system	Integrate weather forecast data for future dates.	**13**
		Provide accurate outfit suggestions based on forecast.	
TS4	Technical Story 4: Build UI for future date outfit planning and customization	Add a user-friendly interface for future date selection.	**8**
		Add interface for customizing outfit preferences.	

PROJECT DEMO

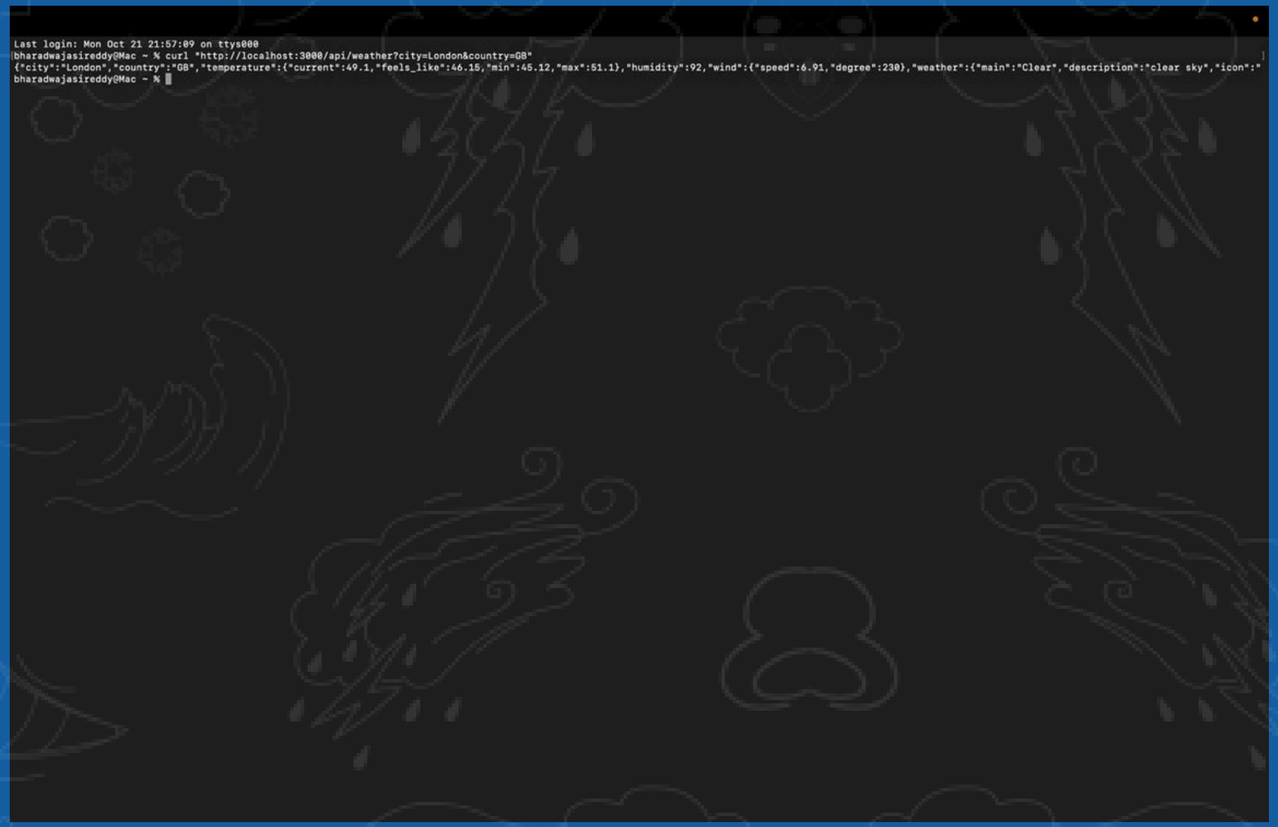
User Story 1 - Typing in Location Manually



App takes Location Using GPS User story 2



API


Last login: Mon Oct 21 21:57:09 on ttys000
bharadwajasireddy@Mac ~ % curl "http://localhost:3000/api/weather?city=London&country=GB"
{"city":"London","country":"GB","temperature":{"current":49.1,"feels_like":46.15,"min":45.12,"max":51.1},"humidity":92,"wind":{"speed":6.91,"degree":230},"weather":{"main":"Clear","description":"clear sky","icon":"
bharadwajasireddy@Mac ~ %

GITHUB LINK

https://github.com/htmw/2024F-Bad-Ideas/wiki

THANK YOU