Deka° - Weather App

CT30A2910 Introduction to Web Programming - Project work

Made by Dekan Ghazi

Introduction

In this course project, I built a weather app called Deka°. The aim of the project is to design a functional, visually appealing and responsive weather application that operates fluidly on both desktop and mobile. The app allows users to view the current weather, hourly forecast, weekly forecast and a 16-day forecast for any city around the world.

To achieve this, I've integrated multiple public APIs such as OpenWeatherMap, WeatherStack and Open-Meteo to combine the strengths of each API. For example, OpenWeatherMap provides high-quality weather icons, WeatherData delivers current better weather details (for example UV) and Open-Meteo was great for 24-hour, 7-day and 16-day forecasts.

The app includes responsive design, making sure that the layout automatically adapts to different screen sizes. Furthermore, I've implemented several user-interactive features such as temperature unit switching. User's can switch between metric, imperial and scientific measurements. Other interactive features that I have included in Deka° are geolocation-based weather detection and a favourites list to see your favourite cities, which can be added to the list with a press of a button. Moreover, I've added buttons for users to view the current temperature in the hottest and coldest city in the world, a feature I've added to make the application show a bit of my curious personality. Using the search feature, will show the user UV data, which is automatically rated in the interface (low, moderate, high etc.). All the error messages are displayed in the interface, which disappear after 5 seconds, so the interface stays clean and looks tidy.

The interface also adjusts its color based on the temperature of the current viewed location. The point of this feature was to make the interface feel more interactive and improve the visualisation of the data. I've also incorporated light mode and dark mode for some extra accessibility and to add some pazzazz to the work. It is good to mention that the WeatherStack API has a limit of 100 requests per month.

My vision for this application was to build a sleek, clean and visually appealing application that is easy and fun to use. I've put a lot of effort into the visual aspect of the application and focused on the small details (you'll notice that in the code) to ensure that my vision is fulfilled.

Points Proposal

Feature	Max points
Well written PDF report	3
Application is responsive and can be used on both desktop and mobile environment	4
Application works on Firefox, Safari, Edge and Chrome	3
The application has clear directory structure and everything is organized well	2
User can search for locations	1
User can use his/her location GPS-coordinates (Geolocation API)	2
At least two data/forecast providers are used (this means completely different data sources like x.com and y.com, not just different API endpoints on same service)	3
At least three data/forecast providers are used	2
User sees the current weather at a specific location	1
User sees the forecast for the next 24 hour, hourly based	3
User sees the forecast for the next 7 days	3
All the weather forecast elements uses icons (and numbers) for e.g. sunny and cloudy weathers	3
The look and feel of the application reflects the current weather (e.g. it is blueish, when it is cold; reddish, when it is hot;, dark, when it is night…)	2
User has the option to tag some locations as her favorites and thus access them from the favorites menu	2
User has an option to switch between celsius and fahrenheit degrees and kelvins	2

My own features:	
16-day extended forecast	2
Light/Dark Mode	2
UV index with rating	2
Hottest and coldest city buttons	1
Error message system	1
Adaptive icon sets for day/night (for forecasts)	1
Interactive navigation bar	2
Weather information displayed (Wind, feels like, humidity, UV)	3
7-day and 16-day forecast shows the high and lowest temperatures	3
Documentation available through app	1
Wind-speed conversion system (if choosing imperial units, you will also get mph instead of km/h)	1
Hovering over forecast feature	1
Added font	1
Visual design of the application	4
Yhteensä:	**61**

Things to keep in mind:
Open-Meteo doesn’t have kelvin temperatures, so it shows celsius instead.

Declaration of AI usage:

AI system used: ChatGPT (OpenAI GPT-5)

Purpose of use	Description of use
Logic for forecastBuilder() and time trim for hourlyForecastBuilder()	AI assisted me in generating the logic for forecastBuilder() and the logic in only getting the hourly time in hourlyForecastBuilder() (the API gave me the whole date, but AI helped me to trim it to only the hour)
AI assisted me with using CSS and mobile responsiveness	I'd like to mention that I used AI in CSS by telling it precisely what I want and asking it me to teach me how to achieve it.
General debugging	AI helped identify typos and logic errors that I couldn't catch myself
Logo and background	AI was used to create the logo and background with my description of my wishes.
updateColor()	The function uses temperatures to determine what color the interface should be changed to. The original code that works for celsius is made by me, but i used chatgpt to make the same method for fahrenheit and kelvin.
emojis in forecast	I used chatgpt to generate me a list of emojis for the forecast icons.
General assistance	Help when stuck or don't know how to proceed