PROJECT REPORT

(Project Term January- May 2023)

(Weather Website)

Submitted by

Name: Akash Sharma **Registration Number : 12219576**

Course Code: INT-222
Course Title: Advanced Web-Development

Submitted To:

Arwinder Dhillon

School of Computer Science and Engineering

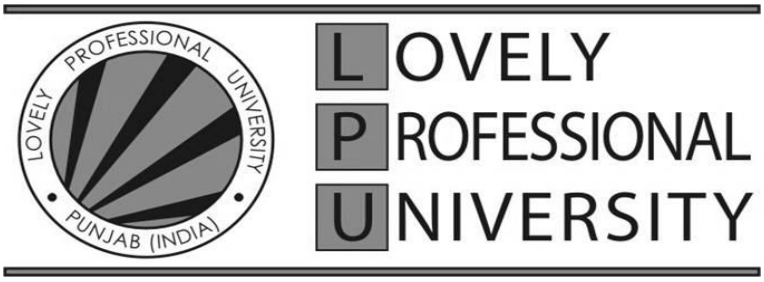

TABLE OF CONTENTS :

DECLARATION :

We hereby declare that the project work entitled (“Weather Website”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in ___CSE____(Programme Name) from Lovely Professional University, Phagwara, under the guidance of (Arwinder Dhillon), during January to May 2023. All the information furnished in this project report is based on my own intensive work and is genuine.

Name of Student : Akash Sharma
Registration Number : 12219576

Date : 15,April,2024

Acknowledgement :

I extend my heartfelt gratitude to all who have contributed to the completion of this project. This project has been a labor of love and dedication, and I am immensely grateful for the opportunity to see it through to fruition. First and foremost, I owe a debt of gratitude to myself for my unwavering commitment, tireless effort, and relentless pursuit of excellence in bringing this project to life. I have poured my heart and soul into every aspect of its development, overcoming challenges and pushing boundaries to achieve success. My journey would not have been possible without the invaluable support and encouragement of my Teacher Arwinder Dhillon whose unwavering belief in me has been a constant source of strength. I am also grateful to my Teacher providing the necessary resources and support for me to embark on this endeavor. Additionally, I extend my thanks to my friends and peers for their unwavering support, constructive feedback, and words of encouragement throughout this journey. Lastly, I acknowledge the countless hours spent by myself in researching, designing, and implementing various aspects of this project. Thank you to everyone who has played a part in this journey, directly or indirectly; your contributions are deeply appreciated and will always be remembered. [Akash Sharma]

Table of Contents……..……

1.Introduction :

Weather influences our daily lives in numerous ways, from planning outdoor activities to making travel decisions. In today's fast-paced world, having access to accurate and up-to-date weather information is essential. With this in mind, the aim of this project is to create a dynamic weather website that seamlessly delivers real-time weather updates to users. Utilizing a combination of HTML, CSS, JavaScript, Bootstrap 4, Node.js, Express.js, and a real-time API, our website will provide users with comprehensive weather data in an intuitive and user-friendly interface.

By harnessing the power of modern web development technologies, we seek to empower users with the information they need to make informed decisions about their day-to-day activities. This project not only demonstrates the capabilities of web development but also serves as a practical solution to a common real-world need. Through this endeavor, we endeavor to create a valuable resource that enhances the lives of users by keeping them informed and prepared for whatever weather comes their way.

2. Profile of the Problem & Rationale

2.1 Need for Accurate Weather Information

In today's fast-paced world, where weather conditions can change rapidly, there is a growing demand for access to accurate and up-to-date weather information. Whether it's planning outdoor activities, scheduling travel arrangements, or simply deciding what to wear for the day, having reliable weather forecasts is essential for making informed decisions. However, existing weather websites often lack real-time updates and may not always provide the most accurate data. Therefore, there is a clear need for a weather website that not only offers comprehensive and precise weather information but also delivers it in real-time.

2.2 Scope: Developing a Real-time Weather Website

The scope of this study encompasses the development of a dynamic weather website that leverages modern web technologies to fetch real-time weather data from an API and present it to users in an intuitive and user-friendly interface. By utilizing HTML, CSS, JavaScript, Bootstrap 4, Node.js,

Express.js, and a real-time API, we aim to create a platform that delivers accurate and up-to-date weather forecasts to users across various devices and platforms. This project will not only address the immediate need for reliable weather information but also serve as a practical demonstration of the capabilities of web development in meeting real-world challenges. Through this endeavor, we strive to provide users with a valuable tool that enhances their daily lives by keeping them informed and prepared for changing weather conditions.

3. Existing System

3.1 Introduction to Existing Weather Websites

Existing weather websites serve as valuable resources for accessing weather forecasts and updates. However, they often face limitations in terms of providing real-time data and user-friendly interfaces. Many of these websites rely on periodic updates, which may not always reflect the latest weather conditions accurately. Additionally, the interfaces of some existing weather websites can be cluttered and overwhelming, making it challenging for users to find the information they need quickly and efficiently.

3.2 Available Weather APIs

Several weather APIs are available for fetching weather data, including OpenWeatherMap, WeatherAPI, and AccuWeather. These APIs offer a range of features, including current weather conditions, hourly forecasts, and weather alerts. By leveraging these APIs, developers can access accurate and up-to-date weather data to integrate into their applications and websites.

3.3 Data Flow Diagram (DFD) of Present System

A simple Data Flow Diagram (DFD) illustrates how weather data flows from the API to the website. At the core of the system is the weather API, which serves as the primary source of weather data. The website interacts with the API to fetch the required weather information, such as current conditions, forecasts, and alerts. The fetched data is then processed and formatted by the website before being displayed to the user in a visually appealing and informative manner.

3.4 Features of the New Weather Website

The new weather website aims to address the limitations of existing systems by introducing several innovative features. One of the key highlights of the new system is its ability to provide real-time updates, ensuring that users always have access to the latest weather information. Additionally, the website will boast a user-friendly interface designed to streamline the user experience and make navigating weather data effortless. Features such as interactive maps, customizable alerts, and personalized forecasts will further enhance the user experience, setting the new weather website apart from its predecessors. Overall, the system to be developed promises to revolutionize the way users access and interact with weather information online, providing a comprehensive and seamless weather forecasting experience.

4. Problem Analysis

4.1 Product Definition: Dynamic Weather Website

The weather website serves as a comprehensive platform for users to access accurate and up-to-date weather information. Key features of the product include real-time weather updates, customizable forecasts, interactive maps, and personalized alerts. Users will have the ability to view current weather conditions, hourly forecasts, and extended forecasts for their location. The website will also provide weather alerts for severe weather events, ensuring users stay informed and prepared. Additionally, the website will feature a user-friendly interface designed to enhance the overall user experience, with intuitive navigation and visually appealing design elements. By offering a wide range of features and functionalities, the weather website aims to become the go-to destination for users seeking reliable weather forecasts.

4.2 Feasibility Analysis

The feasibility of the project hinges on several factors, including technical requirements, availability of APIs, and development resources. In terms of technical requirements, the project requires expertise in web development technologies such as HTML, CSS, JavaScript, Bootstrap 4, Node.js, and Express.js. Additionally, the project relies on access to weather APIs, such as OpenWeatherMap or WeatherAPI, to fetch real-time weather data. Assessing the availability and reliability of these APIs will be crucial in ensuring the project's success. Furthermore, the project requires adequate development resources, including time, manpower, and financial resources, to execute effectively. Conducting a thorough feasibility analysis will help identify any potential

challenges or limitations and devise strategies to mitigate them, thereby increasing the likelihood of project success.

4.3 Project Plan

The project plan outlines the development process for the weather website, including tasks, timelines, and resources needed. The plan begins with a comprehensive analysis of project requirements and objectives, followed by the creation of a detailed project roadmap. Key tasks include designing the website's architecture, developing frontend and backend components, integrating weather APIs, and testing the website for functionality and performance. The project timeline is divided into distinct phases, with specific milestones and deadlines to track progress and ensure timely delivery. Additionally, the project plan allocates resources effectively, including assigning roles and responsibilities to team members and securing necessary tools and technologies. By adhering to the project plan, the development team can streamline the development process and maximize efficiency, ultimately leading to the successful completion of the weather website.

5. Software Requirement Analysis

5.1 Importance of Software Requirement Analysis

Software Requirement Analysis is a crucial part of the software development process. It involves understanding and defining what the software is supposed to accomplish. It sets the foundation for the design, development, and testing phases that follow.

5.2 General Requirements

These are the broad, high-level functionalities that the software should provide. For our weather website, this could include displaying current weather information, providing forecasts, and being responsive on different devices.

5.3 Specific Requirements

These are detailed functionalities that the software should provide. For example, the website should fetch real-time weather data from an API, update the displayed weather information when a user selects a different city, etc.

6. Design

6.1 System Design: Frontend and Backend Architecture

The frontend of our website will be built using HTML, CSS, JavaScript, and Bootstrap 4. The backend, responsible for fetching and processing data from the weather API, will be built using Node.js and Express.js.

6.2 Design Notations

We will use standard design notations like UML diagrams to represent the architecture and flow of our application.

6.3 Detailed Design Specifications

These will provide a detailed view of the system architecture and components. They will include the data flow diagrams, database schema, and more.

6.4 Flowcharts

Flowcharts will be used to visually represent the flow of data and processes in our application.

6.5 Pseudo Code

Pseudo code will be used to represent the logic of our algorithms in a human-friendly format.

7. Testing

7.1 Functional Testing

This involves testing the functionality of our website – for example, whether clicking a button fetches and displays the correct data.

7.2 Structural Testing

This involves testing the internal workings of our application. It includes unit testing individual functions and integration testing the interaction between different parts of our application.

7.3 Levels of Testing

We will perform unit testing, integration testing, system testing, and acceptance testing.

7.4 Testing the Project

We will test our weather website thoroughly to ensure all functionalities work as expected and the user experience is smooth and intuitive.

8. Implementation

8.1 Project Implementation

This involves the actual coding and building of our website, following the design specifications and requirements we have laid out.

8.2 Conversion Plan: Development to Production

After testing, we will move our project from the development phase to the production phase. This involves deploying our website to a server so it can be accessed by users.

8.3 Post-Implementation & Maintenance

After implementation, we will need to maintain our website, fixing any bugs that arise and adding new features as necessary.

Source code : Git Hub