**NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY**



**PROJECT REPORT (KCS -354) Department of Computer Science and Engineering**

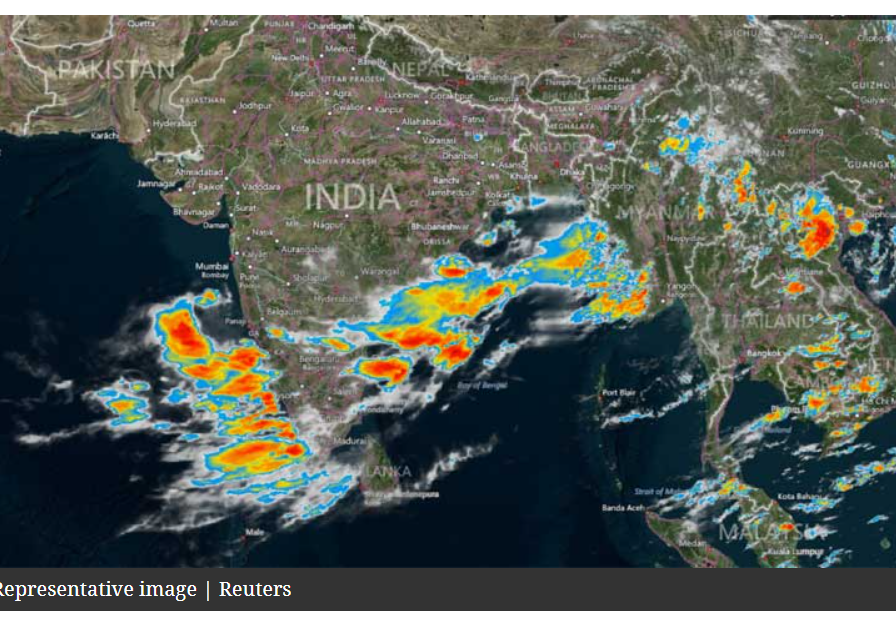
**Submitted by: Submitted to:**

**HASSAN SUHAIL (1901330100121) MS. ANKITA SHARMA**

**MOHAMMAD SHAHID (1901330100155)**

**SYNOPSIs ON**

**Weather forecast website**



**CERTIFICATE**

This is to certify that the Mini Project report entitled “**Weather forecast website**” is a record of the work done by the following student:

STUDENT NAME: ROLL NUMBER:

HASSAN SUHAIL 1901330100121

MOHAMMAD SHAHID 1901330100155

This work is done under our supervision and guidance during the academic year of 2019-20. This report is submitted to the **Noida Institute of Engineering & Technology, Greater Noida** for partial fulfilment for the degree of B.TECH. (Computer Science and Engineering) of **Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India.**

We wish him all the best for all the endeavors

**Signature of Guide:**

**Ms. Ankita Sharma**

**(Assistant professor)**

**ACKNOWLEDGEMENT**

I would like to place on record my deep sense of gratitude to **Ms. Ankita Sharma** (assistant professor) in Department of Computer Science and Engineering, **Noida Institute of Engineering & Technology, Greater Noida,** Gautam Buddha Nagar, Uttar Pradesh, India for his/her generous guidance, help and useful suggestions.

I express my sincere gratitude to **Prof. Chandra Shekhar Yadav, HOD(CSE),** Noida Institute of Engineering & Technology, Greater Noida for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

**Date: Student Name:**

***24TH JAN,2021 HASSAN SUHAIL (1901330100121)***

***MOHAMMAD SHAHID (1901330100155)***

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Topic** | **Signature/Remarks** |
| **1.** | Introduction |  |
| **2.** | Objective |  |
| **3.** | Software Required |  |
| **4.** | Literature Review |  |
| **5.** | Methodology |  |
| **6.** | Plan of Work |  |
| **7.** | Modules |  |
| **8.** | Tools and Techniques Required |  |
| **9.** | Project Design Phase |  |
| **10.** | Data Flow Diagram |  |
| **11.** | Project snapshot |  |
| **12.** | Coding |  |
| **13.** | References |  |

**INTRODUCTION**

Weather forecasting is the application of science and technology to predict the conditions of the atmosphere for a given location and time. People have attempted to predict the weather informally for millennia and formally since the 19th century. Weather forecasts are made by collecting quantitative data about the current state of the atmosphere at a given place and using meteorology to project how the atmosphere will change.

Once calculated by hand based mainly upon changes in barometric pressure, current weather conditions, and sky condition or cloud cover, weather forecasting now relies on computer-based models that take many atmospheric factors into account.[1] Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, teleconnections, knowledge of model performance, and knowledge of model biases. The inaccuracy of forecasting is due to the chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric processes. Hence, forecasts become less accurate as the difference between current time and the time for which the forecast is being made (the range of the forecast) increases. The use of ensembles and model consensus help narrow the error and pick the most likely outcome.

**OBJECTIVE**

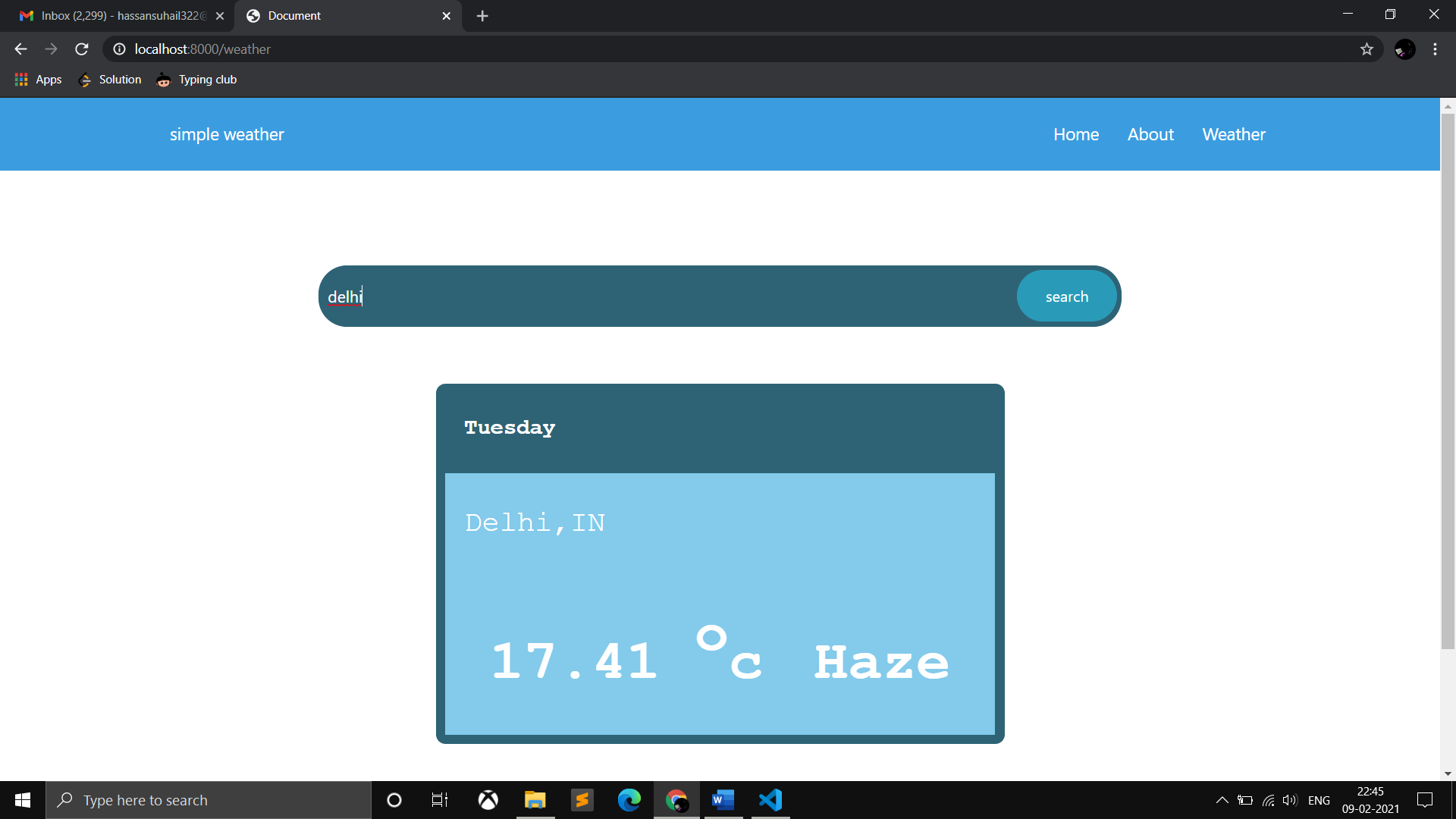
The goal of weather prediction is to provide information people and organizations can use to reduce weather-related losses and enhance societal benefits, including protection of life and property, public health and safety, and support of economic prosperity and quality of life.

**PROJECT BENEFITS**

* We can check weather at any time
* Farmers can known when to plant or harvest their crops
* People can choose where and when to take their holidays to take advantages of good weather
* Surfers known when large waves are expected
* Regions can be evacuated if hurricanes or floods are expected.

**SOFTWARE REQUIRED**

* ***FRONT-END: HTML,CSS,JAVASCRIPT***



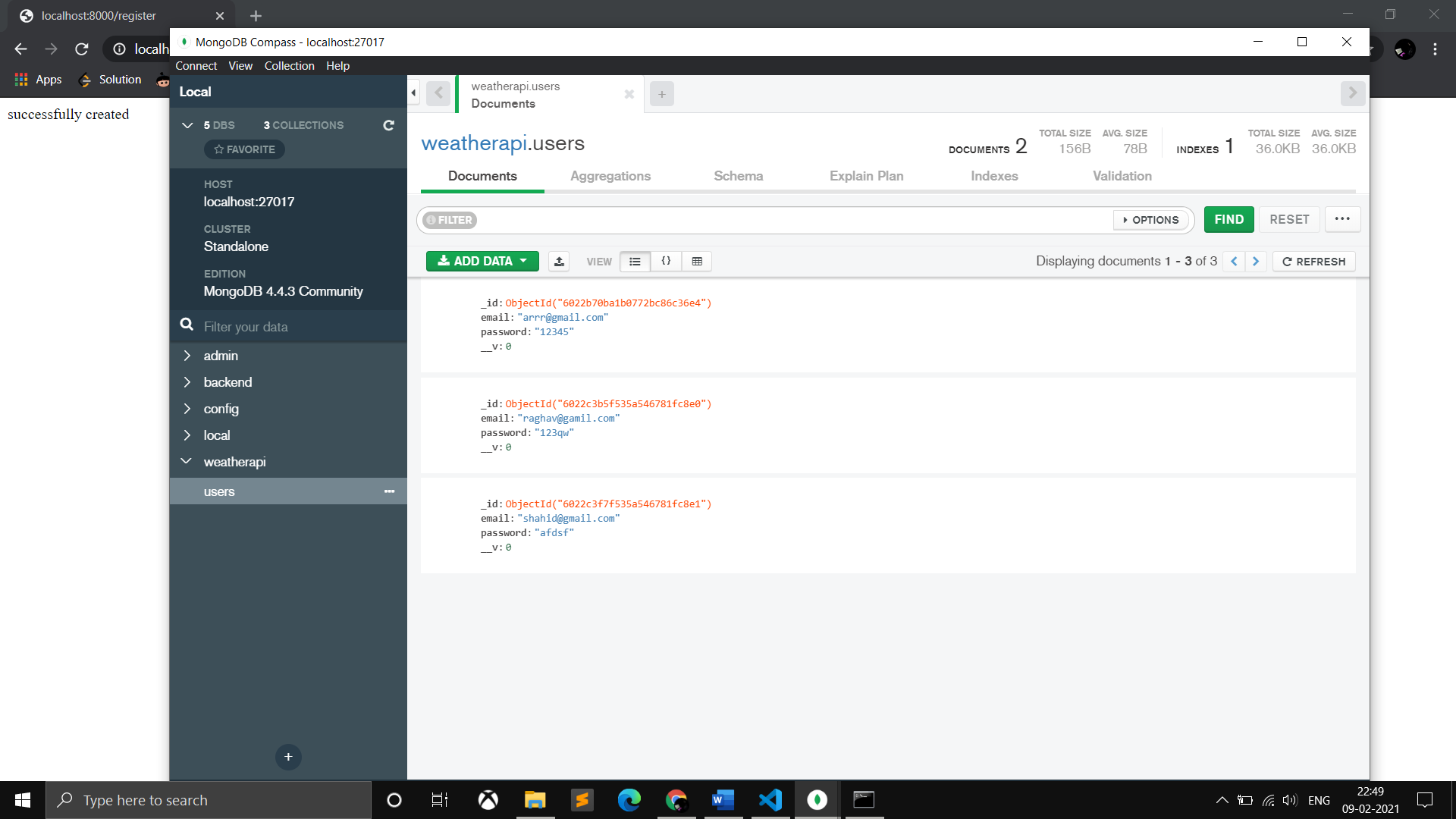
HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.[10] JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but they are now core components of other runtime systems, such as Node.js and Deno. These systems are used to build servers and are also integrated into frameworks, such as Electron and Cordova, for creating a variety of applications.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

***BACK-END : MONGO DB***

***MONGO DB:***

**MongoDB** is a [source-available](https://en.wikipedia.org/wiki/Source-available) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [document-oriented database](https://en.wikipedia.org/wiki/Document-oriented_database) program. Classified as a [NoSQL](https://en.wikipedia.org/wiki/NoSQL) database program, MongoDB uses [JSON](https://en.wikipedia.org/wiki/JSON)-like documents with optional [schemas](https://en.wikipedia.org/wiki/Database_schema). MongoDB is developed by [MongoDB Inc.](https://en.wikipedia.org/wiki/MongoDB_Inc.) and licensed under the [Server Side](https://en.wikipedia.org/wiki/Server_Side_Public_License)

10gen software company began developing MongoDB in 2007 as a component of a planned platform as a service product. In 2009, the company shifted to an open-source development model, with the company offering commercial support and other services. In 2013, 10gen changed its name to MongoDB Inc.

* **FEATURES:**
* High Performance.
* High Availability.
* Scalability and Flexibility Run anytime..
* Web and Data Warehouse Strengths.
* Strong Data Protection.

**LITERATURE REVIEW**

We see and Interact with various types of weather forecasting websites and come up with the idea of creating our own website which is Simple, Reliable and has minimalistic design so that users can interact with a piece of mind. It can Keep the information of current weather of any place. The Existing part of this project is; it display real time weather of any place around the globe.

This website uses an API (application programming interface) which provides the real time data of the weather.

**HIGLIGHTS OF PROJECTS:**

* You can login to this website
* You can check the weather any time

**METHODOLOGY**

**SOFTWARE DEVELOPMNET LIFE CYCLE(SDLC)**

System Development Life Cycle consists of two major steps of System analysis and design. It includes set of activities that analysis, designers and users carry out to develop and implement an information system.

|  |
| --- |
| **FEASIBILITY STUDY** |

|  |
| --- |
| **REQUIREMENT ANALYSIS AND SPECIFICATION** |

|  |
| --- |
| **DESIGN** |

|  |
| --- |
| **INTEGRATION AND SYSTEM TESTING** |

|  |
| --- |
| **CODING AND UNIT TESTING** |

|  |
| --- |
| **MAINTENANCE** |

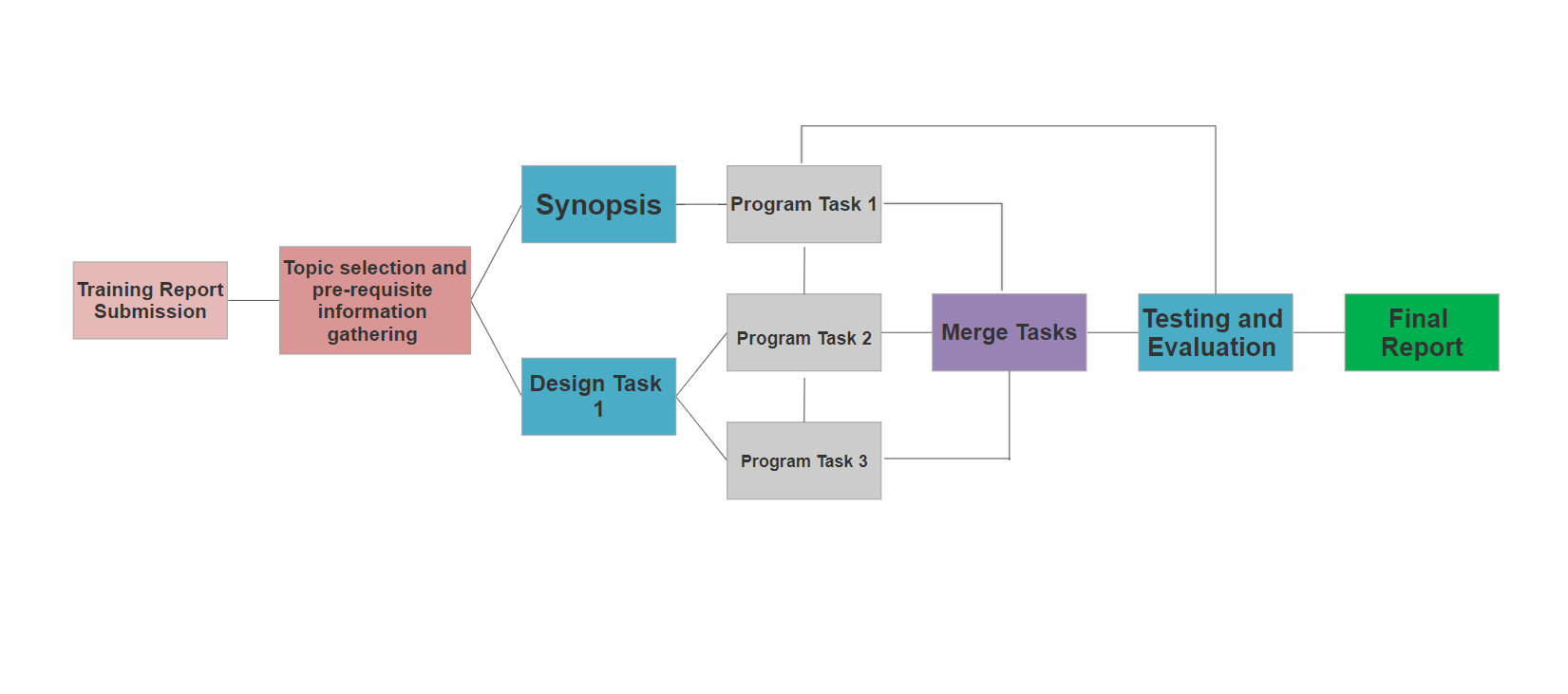
**PLAN OF WORK**

When errors are detected at some later phase, these feedback paths allow correcting errors committed by programmers during some phase. The feedback paths allow the phase to be reworked in which errors are committed and these changes are reflected in the later phases.

This website uses the API which is provided by openweatherapi which is a API provider, it gives the current time weather information

The systems development life cycle consists of the following activities:

1. Preliminary Investigation.
2. Determination Requirements.
3. Design of System.
4. Development of System.
5. System Testing.
6. Implementation.

**PERT CHART**

* Starting from information gathering about the weather forecasting website.
* Designing the layout .
* Developing it to code.
* Finally building the website .

**GANTT CHART**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Project Lead: HASSAN SUHAIL**  **(1901330100121)** | | **MONTH:** | **NOVEMBER,**  **2020** | **DECEMBER,2020** | | | | **JANUARY,2021** | | | |
|  | **DAYS:** | **30** | **5** | **12** | **19** | **26** | **3** | **10** | **17** | **24** |
| **Sno.** | **TASK** | **START**  **DATE** | **END**  **DATE** |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **1.** | **Training Report**  **Submission** | 30, Nov | 5, Dec |  | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2.** | **Training Report**  **Presentation** | 5, Dec | 12, Dec |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3.** | **Pre-Requisite**  **Information Gathering** | 5, Nov | 12, Dec |  |  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4.** | **Synopsis** | 26, Dec | 3, Jan |  |  |  |  |  | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5.** | **Design** | 12, Dec | 3, Jan |  |  |  | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **6.** | **Development/**  **Coding** | 12, Dec | 10, Jan |  |  |  | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7.** | **Testing and**  **Evaluation** | 26, Dec | 24, Jan |  |  |  |  |  | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8.** | **Final Report** | 17, Dec | 24, Jan |  |  |  |  |  |  |  |  | |

**Training Report**  **Synopsis Project**

**Submission Report**

**Submission**

**MODULES**

The Module Descriptions of weather forecast Project. These Modules will be developed

**1. Register:** A user who does not have an account can make with this module.

**2. Login:** Account Holders can login in to the system using this module.

**3. Logout:** Users can logout using this module

**3. Input:** This module takes the name of the city which is entered by the user and fetch the data regarding this city.

**4. Display:** This module displays the real time weather of the entered city

**5. search:** This module fetches the data from the api and compare it with the entered name and then send it to display

**TOOLS AND TECHNIQUES REQUIREMENTS**

1. **Hardware Requirements Specification**

**● Processor**  : Intel Pentium III or later

**● Main Memory (RAM)**  : 8 GB

**● Cache Memory**  : 512 KB

**● Monitor** :14 Inch Color Monitor

**● Keyboard**  : 108 Keys

**● Mouse** : Optical Mouse

**● Hard Disk** : 512 GB

1. **Software Requirements Specification** DB

**● Front End/Language** : HTML,CSS,JAVASCRIPT

**● Back End/Database** : NODE JS,MONGO

**● Additional Tools** :MONGO DB COMPASS,VS CODE

**● Operating System** : Windows 7, 8, 9, 10, XP

**CODE PHASE**

**LANDING PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

    {{>headlinks}}

</head>

<body>

    <div class="container-fluid header  ">

        <div class="row">

            <div class="col-md-10 col-12 mx-auto mt-5 mainpage ">

                <h1>Welcome to this website please login or register to check the current weather of your city</h1>

                <div class="logout">

                    <a href="/login" class="btn btn-primary">Login</a>

                    <a href="/register" class="btn btn-danger">Register</a>

                </div>

            </div>

        </div>

    </div>

</body>

</html>

**LOGIN PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

    {{>headlinks}}

</head>

<body>

    <div class="container mt-5">

        <a href="/" class="btn btn-primary mb-5">Go Back</a>

        <h1>Login</h1>

        <form action="/login" method="POST">

            <div class="form-group">

                <label for="exampleInputEmail1">Email address</label>

                <input type="email" name="email" class="form-control" id="exampleInputEmail1"

                    aria-describedby="emailHelp">

                <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone

                    else.</small>

            </div>

            <div class="form-group">

                <label for="exampleInputPassword1">Password</label>

                <input type="password" name="password" class="form-control" id="exampleInputPassword1">

            </div>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

    </div>

</body>

</html>

**HOME PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

  {{>headlinks}}

</head>

<body>

  {{>navbar}}

  <section>

    <div class="container-fluid header">

      <div class="row">

        <div class="col-md-10 col-12 mx-auto mt-5">

          <div class="row">

            <div class="col-md-6 col-12 mx-auto header-left  ">

              <p>welcome to this webliste</p>

              <h1>get the current <span class="txt">weather</span> info of your city</h1>

              <a href="/weather" <button class="btn">click here</button></a>

            </div>

            <div class="col-md-6 col-12 mx-auto header-right">

              <div id="carouselExampleControls" class="carousel slide" data-ride="carousel">

                <div class="carousel-inner">

                  <div class="carousel-item active">

                    <img class="d-block w-100" src="./img/K006.jpg" alt="First slide">

                  </div>

                  <div class="carousel-item">

                    <img class="d-block w-100" src="./img/4.jpg" alt="Second slide">

                  </div>

                  <div class="carousel-item">

                    <img class="d-block w-100" src="./img/5.jpg" alt="Third slide">

                  </div>

                </div>

                <a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">

                  <span class="carousel-control-prev-icon" aria-hidden="true"></span>

                  <span class="sr-only">Previous</span>

                </a>

                <a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">

                  <span class="carousel-control-next-icon" aria-hidden="true"></span>

                  <span class="sr-only">Next</span>

                </a>

              </div>

            </div>

          </div>

        </div>

      </div>

    </div>

  </section>

  {{>footer}}

</body>

</html>

**ABOUT PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

 {{>headlinks}}

</head>

<body>

 {{>navbar}}

  <section>

    <div class="container-fluid header">

      <div class="row">

        <div class="col-md-10 col-12 mx-auto mt-5">

          <div class="row">

            <div class="col-md-6 col-12 mx-auto header-right  ">

          <img src="./img/10.jpg" alt="img not found" width="500" height="500" class="img-fluid">

        </div>

        <div class="col-md-6 col-12 mx-auto header-left">

          <p>welcome to this website</p>

          <h1>get the current   <span class="txt">weather</span> info of your city</h1>

            <a href="/weather"<button class="btn">click here</button></a>

        </div>

      </div>

    </div>

    </div>

    </div>

  </section>

 {{>footer}}

</body>

</html>

**REGISTER**

<!DOCTYPE html>

<html lang="en">

<head>

  {{>headlinks}}

</head>

<body>

   <div class="container mt-5">

    <a href="/" class="btn btn-primary mb-5">Go Back</a>

    <h1>Register Account</h1>

    <form action="/register" method="POST">

        <div class="form-group">

            <label for="exampleInputEmail1">Email address</label>

            <input type="email" name="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">

            <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>

        </div>

        <div class="form-group">

            <label for="exampleInputPassword1">Password</label>

            <input type="password" name="password" class="form-control" id="exampleInputPassword1">

        </div>

        <button type="submit" class="btn btn-primary">Register</button>

    </form>

</div>

</body>

</html>

**WEATHER PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

  {{>headlinks}}

</head>

<body>

  {{>navbar}}

  <div class="container-fluid main\_header">

    <div class="row">

      <div class="col-md-10 col-12 mx-auto">

        <div class="main\_content">

          <form class="temp\_form">

            <input class=" no" type="text " id="cityName" placeholder="enter your city name" autocomplete="off">

            <input  type="submit" value="search" id="submitbtn">

          </form>

        </div>

        <div class="tempInformation">

          <div class="top\_layer">

            <p id="day"></p>

            {{!-- <p id="today\_data">23 jan </p> --}}

          </div>

          <div class="main\_layer">

            <p id="city\_name">Get output here</p>

            <div class="middle\_layer data\_hide">

              <p id="temp"> <span id="temp\_real"> 0 </span> <sup>o</sup>c </p>

              <p id="temp\_dis">hi</p>

            </div>

          </div>

        </div>

      </div>

    </div>

  </div>

  {{>footer}}

  <script src="js/main.js"></script>

  <script>

    const getCurrentDay = () => {

      let weekday = new Array(7);

      weekday[0] = "sunday";

      weekday[1] = "Monday";

      weekday[2] = "Tuesday";

      weekday[3] = "Wednesday";

      weekday[4] = "Thursday";

      weekday[5] = "Friday";

      weekday[6] = "Saturday";

      let currentTime = new Date();

      days = weekday[currentTime.getDay()];

      const day = document.getElementById('day');

      day.innerText = days;

    };

    getCurrentDay();

  </script>

</body>

</html>

**FOOTER**

  <div class="contaiver-flex">

    <div class="row">

      <div class="col-md-12 col-12 mx-auto f">

        <p>created by Hasan and shahid</p>

      </div>

    </div>

  </div>

**HEADER LINKS**

 <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

  <!-- jQuery library -->

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <!-- Popper JS -->

  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

  <!-- Latest compiled JavaScript -->

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

  <title>Document</title>

  <link rel="stylesheet" href="css/style.css">

**NAVBAR**

 <div class="container-fluid main\_menu">

    <div class="row">

      <div class="col-md-10 col-12 mx-auto">

        <nav class="navbar navbar-expand-lg  ">

          <a class="navbar-brand" href="#">simple weather</a>

          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"

            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

          </button>

          <div class="collapse navbar-collapse" id="navbarSupportedContent">

            <ul class="navbar-nav ml-auto">

              <li class="nav-item active">

                <a class="nav-link" href="/index">Home <span class="sr-only">(current)</span></a>

              </li>

              <li class="nav-item active">

                <a class="nav-link" href="/about">About <span class="sr-only">(current)</span></a>

              </li>

              <li class="nav-item">

                <a class="nav-link" href="/weather">Weather</a>

              </li>

              <li class="nav-item">

                <a class="nav-link" href="/logout">Logout</a>

              </li>

            </ul>

          </div>

        </nav>

      </div>

    </div>

  </div>

**BACKEND /NODE JS**

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

require('./db/conn');

const User = require("./models/user");

const path = require('path')

const hbs = require('hbs');

const user = require('./models/user');

const app = express();

const port = process.env.PORT || 8000;

const static\_path = path.join(\_\_dirname, '../public');

const views\_path = path.join(\_\_dirname, '../templates/views');

const partials\_path = path.join(\_\_dirname, "../templates/partials");

app.set('view engine', 'hbs');

app.set('views', views\_path);

hbs.registerPartials(partials\_path);

app.use(bodyParser.urlencoded({extended: true}));

app.use(express.static(static\_path));

app.get('/', (req, res) => {

    res.render('main');

});

app.get('/register', (req, res) => {

    res.render('register');

});

app.get('/login', (req, res) => {

    res.render('login');

});

app.get('/Logout', (req, res) => {

    res.render('main')

});

app.get('/weather', (req, res) => {

    res.render('weather')

});

app.get('/about', (req, res) => {

    res.render('about')

});

app.get('/index', (req, res) => {

    res.render('index')

});

// app.get('/index', (req, res) => {

//     res.render('index')

// });

// app.get('/about', (req, res) => {

//     res.render('about')

// });

// app.get('/weather', (req, res) => {

//     res.render('weather')

// });

// app.get('\*', (req, res) => {

//     res.render('err');

// });

// POST REQUESTS

app.post('/register',(req,res) => {

    const email = req.body.email;

    const password = req.body.password;

    const newUser = new User({

        email : email,

        password : password

    });

    newUser.save((err) =>{

        err?console.log(err): res.send('successfully created');

    })

})

app.post('/login',(req,res) =>{

    const email = req.body.email;

    const password = req.body.password;

    User.findOne({email : email}, (err,foundResults) =>{

        if(err){

            console.log(err);

        }else{

            if(foundResults.password === password){

                res.render('index');

            }else{

                res.send('incorrect email or password');

            }

        }

    })

})

app.listen(port, () => {

    console.log('it is working successfully..')

});

**JAVA SCRIPT**

const submibtn = document.getElementById('submitbtn');

const cityName = document.getElementById('cityName');

const city\_name = document.getElementById('city\_name');

const temp\_real = document.getElementById('temp\_real');

const temp\_dis = document.getElementById('temp\_dis');

const datahide = document.querySelector(".middle\_layer");

const getInfo = async (event) => {

    event.preventDefault();

    let cityVal = cityName.value;

    if (cityVal === "") {

        city\_name.innerText = `Please enter name before search`;

        datahide.classList.add('data\_hide');

    } else {

        try {

            let url = `http://api.openweathermap.org/data/2.5/weather?q=${cityVal}&units=metric&appid=afc0d846ef746a4a98cb980be68b0f06`

            const response = await fetch(url);

            const data = await response.json();

            const arrData = [data];

            city\_name.innerText = `${arrData[0].name},${arrData[0].sys.country}`;

            temp\_real.innerText = arrData[0].main.temp;

            temp\_dis.innerText = arrData[0].weather[0].main;

            datahide.classList.remove('data\_hide');

        } catch {

            city\_name.innerText = `Please enter name properly`;

            datahide.classList.add('data\_hide');

        }

    }

}

submibtn.addEventListener('click', getInfo);

**CSS**

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

html{

    font-size: 62.5%;

}

.main\_menu{

    background-color:#3b9de0;

}

.row{

    margin: 0;

}

                                 /\* Navbar style \*/

.navbar{

    background-color:#3b9de0;

    margin : 1.5rem;

}

.navbar a{

    color: white;

    font-size: 1.8rem;

}

.navbar a:hover{

    color: rgb(11, 54, 102);

}

.navbar li {

    margin-left: 2rem;

}

.navbar-toggler-icon{

    background-color: #1c262cc7;

    color: rgb(238, 238, 231);

}

                                   /\* Header left \*/

.header-left{

 height: 60rem;

 display: flex;

 flex-direction: column;

 justify-content: center;

 text-transform: capitalize;

 align-items: flex-start;

}

.header-left h1{

    font-size: 2.6rem;

    margin:1rem 0 5rem 0;

}

.header-left p {

    font-size: 2.6rem;

}

.header-left .txt{

    color: rgb(75, 174, 212)

}

.btn{

    outline: none;

    background: rgb(50, 177, 228);

    border-radius: 3rem;

    padding: 0.6rem 1rem;

    color: white;

    font-size: 1.6rem;

}

.btn :hover{

    filter: drop-shadow(drop-shadow(0 0.5rem 1rem black));

}

                        /\* Header right in which we have crousal \*/

.header-right{

    height: 60rem;

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    justify-content: center;

}

                            /\* Styling temp form  \*/

.temp\_form{

    width: 70%;

    position: relative;

    margin: 10rem auto 6rem;

    background-color : #2e6376;

    border-radius: 4rem;

}

.no{

    width: 100%;

    padding: 2rem 5rem 2rem 1rem;

    color:white;

    background-color: #2e6376;

    outline: none;

    border: none;

    font-size: 1.7rem;

    border-radius: 3rem;

}

::placeholder{

    color:white;

}

.temp\_form input[type="submit"]{

    position: absolute;

    top: 5px;

    right: 5px;

    bottom: 5px;

    border: none;

    outline: none;

    background: rgb(41, 155, 184);

    border-radius: 3rem;

    padding: 0 3rem;

    color: white;

    font-size: 1.6rem;

}

/\* .temp\_form input [type="submit"] :hover{

    background-color: rgb(12, 12, 11);

    color: red;

} \*/

#submitbtn  :hover{

    color: red;

}

.tempInformation{

    width: 60rem;

    margin: auto;

    min-height: 20rem;

    background-color: #2e6376;

    border-radius: 1rem;

    padding: 1rem;

    font-family: 'Courier New', Courier, monospace;

}

.top\_layer{

    width: 100%;

    height: 20%;

    background-color: 2e6376;

    display: flex;

    justify-content: space-between;

    align-items: center;

    border-radius: 1rem;

}

.top\_layer p{

    color: white;

    font-weight: 600;

    padding: 2rem;

    font-size: 2.3rem;

}

.main\_layer{

    width: 100%;

    height: 80%;

    display: flex;

    background-color: rgb(131, 202, 235);

    flex-direction: column;

    padding: 2rem;

}

.main\_layer #city\_name{

    padding: 1rem 0;

    color:white;

    font-weight: 500;

    font-size: 3.1rem;

}

.main\_layer .middle\_layer {

    display: flex;

    margin-top: 6rem;

    justify-content: space-around;

    color:white;

}

.main\_layer .middle\_layer p{

    font-size: 6rem;

    font-weight: 900;

}

sup{

    color:white;

    font-size: 6rem;

    margin-top: -2rem;

}

.data\_hide{

    visibility:hidden;

    transition: all 0.3s linear;

}

                         /\* styling bottom. \*/

  .f {

    display: flex;

    position: sticky;

    left: 0;

    bottom: 0;

    width: 100%;

    height: 5rem;

    margin-top: 15rem;

    background-color:#3b9de0;

    color: white;

    text-align: center;

    align-items: center;

    justify-content: center;

  }

  .f p{

      text-transform: capitalize;

      font-size: 2rem;

  }

.carousel-item img{

    height: 40rem;

}

.header-right img{

    filter: drop-shadow(0 0.5rem 1rem black);

}

.mainpage{

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

}

.logout{

    margin-top: 10rem;

}

@media (max-width:768px){

    .navbar-brand{

        margin-left:2rem ;

    }

    .header-left{

        height: auto;

        margin: 5rem 0;

    }

    .header-right{

        height: auto;

        margin-top: 5rem ;

    }

    .tempInformation{

        width: auto;

    }

  }

**FRONT END**

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

html{

    font-size: 62.5%;

}

.main\_menu{

    background-color:#3b9de0;

}

.row{

    margin: 0;

}

                                 /\* Navbar style \*/

.navbar{

    background-color:#3b9de0;

    margin : 1.5rem;

}

.navbar a{

    color: white;

    font-size: 1.8rem;

}

.navbar a:hover{

    color: rgb(11, 54, 102);

}

.navbar li {

    margin-left: 2rem;

}

.navbar-toggler-icon{

    background-color: #1c262cc7;

    color: rgb(238, 238, 231);

}

                                   /\* Header left \*/

.header-left{

 height: 60rem;

 display: flex;

 flex-direction: column;

 justify-content: center;

 text-transform: capitalize;

 align-items: flex-start;

}

.header-left h1{

    font-size: 2.6rem;

    margin:1rem 0 5rem 0;

}

.header-left p {

    font-size: 2.6rem;

}

.header-left .txt{

    color: rgb(75, 174, 212)

}

.btn{

    outline: none;

    background: rgb(50, 177, 228);

    border-radius: 3rem;

    padding: 0.6rem 1rem;

    color: white;

    font-size: 1.6rem;

}

.btn :hover{

    filter: drop-shadow(drop-shadow(0 0.5rem 1rem black));

}

                        /\* Header right in which we have crousal \*/

.header-right{

    height: 60rem;

    display: flex;

    flex-direction: column;

    align-items: flex-start;

    justify-content: center;

}

                            /\* Styling temp form  \*/

.temp\_form{

    width: 70%;

    position: relative;

    margin: 10rem auto 6rem;

    background-color : #2e6376;

    border-radius: 4rem;

}

.no{

    width: 100%;

    padding: 2rem 5rem 2rem 1rem;

    color:white;

    background-color: #2e6376;

    outline: none;

    border: none;

    font-size: 1.7rem;

    border-radius: 3rem;

}

::placeholder{

    color:white;

}

.temp\_form input[type="submit"]{

    position: absolute;

    top: 5px;

    right: 5px;

    bottom: 5px;

    border: none;

    outline: none;

    background: rgb(41, 155, 184);

    border-radius: 3rem;

    padding: 0 3rem;

    color: white;

    font-size: 1.6rem;

}

/\* .temp\_form input [type="submit"] :hover{

    background-color: rgb(12, 12, 11);

    color: red;

} \*/

#submitbtn  :hover{

    color: red;

}

.tempInformation{

    width: 60rem;

    margin: auto;

    min-height: 20rem;

    background-color: #2e6376;

    border-radius: 1rem;

    padding: 1rem;

    font-family: 'Courier New', Courier, monospace;

}

.top\_layer{

    width: 100%;

    height: 20%;

    background-color: 2e6376;

    display: flex;

    justify-content: space-between;

    align-items: center;

    border-radius: 1rem;

}

.top\_layer p{

    color: white;

    font-weight: 600;

    padding: 2rem;

    font-size: 2.3rem;

}

.main\_layer{

    width: 100%;

    height: 80%;

    display: flex;

    background-color: rgb(131, 202, 235);

    flex-direction: column;

    padding: 2rem;

}

.main\_layer #city\_name{

    padding: 1rem 0;

    color:white;

    font-weight: 500;

    font-size: 3.1rem;

}

.main\_layer .middle\_layer {

    display: flex;

    margin-top: 6rem;

    justify-content: space-around;

    color:white;

}

.main\_layer .middle\_layer p{

    font-size: 6rem;

    font-weight: 900;

}

sup{

    color:white;

    font-size: 6rem;

    margin-top: -2rem;

}

.data\_hide{

    visibility:hidden;

    transition: all 0.3s linear;

}

                         /\* styling bottom. \*/

  .f {

    display: flex;

    position: sticky;

    left: 0;

    bottom: 0;

    width: 100%;

    height: 5rem;

    margin-top: 15rem;

    background-color:#3b9de0;

    color: white;

    text-align: center;

    align-items: center;

    justify-content: center;

  }

  .f p{

      text-transform: capitalize;

      font-size: 2rem;

  }

.carousel-item img{

    height: 40rem;

}

.header-right img{

    filter: drop-shadow(0 0.5rem 1rem black);

}

.mainpage{

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

}

.logout{

    margin-top: 10rem;

}

@media (max-width:768px){

    .navbar-brand{

        margin-left:2rem ;

    }

    .header-left{

        height: auto;

        margin: 5rem 0;

    }

    .header-right{

        height: auto;

        margin-top: 5rem ;

    }

    .tempInformation{

        width: auto;

    }

  }

**BIBLOGRAPHY**

This document contains provisions which, through reference in this text, constitute provisions of the present document.

* [www.google.com](http://www.google.com) for various searching
* HTML5- Black Book
* [www.getbootstrap.com](http://www.getbootstrap.com)
* Keeves PHP- MYSQL
* [www.stackoverflow.com](http://www.stackoverflow.com)
* Online at [www.w3schools.org](http://www.w3schools.org)
* Worx Press – Beginning PHP5
* Research Paper on [www.scholar.google.com](http://www.scholar.google.com)