VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Md Mohiuddin Emon

ALERT SYSTEM FOR COASTAL AREA OF BANGLADESH

Information Technology
2022

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Smail Menani, my supervisor, for his unwavering support, guidance, and important feedback throughout my thesis. This endeavour would not have been accomplished without his constant supervision and concern.

I would like to extend my gratitude to the Rimes Company and Mr. Raihanul Haque Khan for allowing me to work on this project and providing me with the opportunity to learn many things from it.

I am really grateful to VAMK for providing me with all of the required resources, as well as to the other VAMK teachers who worked diligently to inculcate knowledge and love of studying in us.

Finally, I express my heartfelt gratitude to my companions and family for their never-ending support, encouragement, and attention. They never lose faith in me and motivate me to do well.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Md Mohiuddin Emon
Title	Alert System for Coastal Area Of Bangladesh
Year	2022
Language	English
Pages	34
Name of Supervisor	Smail Menani

The aim of this thesis is to create a web application which will update weather forecast for Coxsbazar and generate alert. In addition, this thesis covers how weather forecasts are interpreted, how alerts are created and how the users will benefit from these alerts.

Before creating this web application, a preliminary design was created and after a thorough analysis technologies were selected. Angular was used as the framework with highcharts as secure data visualisation library. For mapping leaflet with OpenStreetMap was used.

After the implementation of this analysis the aim of this thesis was met. The results of this thesis will make life easier for individuals living around the coast by providing five days of weather forecasting alerts. One of the most important benefits of this web application is that users will be aware of inclement weather and will be able to mitigate harm.

Keywords **I**nterpret, preserve, web Application, and forecast alerts

Contents

LIST OF FIGURES

LIST OF CODE SNIPPETS

1. INTRODUCTION

Every year people in Bangladesh fall victim to natural calamities such as floods, heavy rainfall, and earth quacks. This affects the normal lives of the people. Natural calamities hit Bangladesh because of climate change and many projects have been implemented for controlling the calamities. People need to be alerted so that they take proper precautions. For an alert generation, weather forecasts are important; upcoming rainfall, cold wave, and heatwave can be known from them.

The thesis is basically about Cox's Bazar weather forecasting and generating an alert. An alert is needed because rainfall, heatwave, or cold wave affects people's lives. Most of the people in the villages of Bangladesh are engaged in agriculture, they produce paddy, wheat, and grains and live their livelihood. Sometimes due to heavy rainfall, many crops are wasted, and cattle are harmed. It also affects coastal areas and makes life miserable; in coastal areas floods, tidal surges, and rainfall are the most common occurrences. Mainly in coastal areas, whenever a flood occurs, people are evacuated on an emergency basis, and during this period many people may die. Every year cyclones and tidal surges occur in Bangladesh and there is a lot of damage to houses, crops, and lives. At the end of 2017, when Rohingya issues took place, the Bangladesh Government and the UN nations decided to work together to help them. The sweltering weather has disrupted people's daily lives in the city and its environs, particularly low-income and working-class residents.

The aim of the thesis is about generating alerts through a web application to help the people who are living Cox's Bazar. To develop this application, it is important to analyze the requirement, make the list of required technology and develop the web application and create the alert. Basically, the web application is based on five days of weather forecast and it creates an alert with the help of data. The data is collected from the Bangladesh Meteorological Department. Rainfall, wind gust, heatwaves, and cold waves are known from the weather forecasts. The objectives are to empower and induce individuals, and communities to be prepared for

surges and take action to increment security and diminish harm. If the people know about extreme weather conditions five days earlier, losses can be reduced.

2. THEORITICAL BACKGROUND

Climate is the average weather status in a particular area, where weather shows a certain pattern over the long period of time. The climate of an area includes different data types, such as average temperature in different seasons, sunshine amount, precipitation, rainfall and so on. According to World Meteorological Organization (WMO), the typical period used for defining a climate change is some 30 years[1]. Notwithstanding the uncertainties, climate scientists have reached a strong consensus that in the absence of counter measurements for the climate-change, the changes will be substantial, with long-lasting effects on many of Earth's physical and biological systems.[2] Some examples to understand how climate change can affect the different areas of our lives is given next.

When water warms up, it expands. At the same time, global warming causes polar ice sheets and glaciers to melt. The combination of these changes is causing sea levels to rise, resulting in flooding and erosion of coastal and low-lying areas. Heavy rain and other extreme weather events are becoming more frequent. This can lead to floods and decreasing water quality, but also decreasing availability of water resources in some regions. [3]

Many poor developing countries are among the most affected. People living there often depend heavily on their natural environment**,** and they have the least resources to cope with the changing climate.[4] There has been an increase in the number of heat-related deaths in some regions and a decrease in cold-related deaths in others. We are already seeing changes in the distribution of some water-borne illnesses and disease vectors.

Damage to property and infrastructure and to human health imposes heavy costs on society and the economy. Between 1980 and 2011 floods affected more than 5.5 million people and caused direct economic losses of more than €90 billion. [5]

Higher temperatures can cause increased mortality, reduced productivity and damage to infrastructure. The most vulnerable members of the population, such

as the elderly and infants, will be most severely affected. Higher temperatures are also expected to cause a shift in the geographical distribution of climate zones.

Increased rainfall over extended periods will mainly lead to fluvial (river) flooding, while short, intense cloudbursts can cause pluvial floods, where extreme rainfall causes flooding without any body of water overflowing. As the climate heats up, rainfall patterns change, evaporation increases, glaciers melt and sea levels rise. All these factors affect the availability of fresh water.

3. CASE STUDY

As we can see from the previous section, climate change plays a vital role in basically every aspect of our lives. In our case study, we focused on Bangladesh's climate change and how it can affect the people in Bangladesh.

Bangladesh has a tropical monsoon-type climate, with a hot and rainy summer and a dry winter. January is the coolest month with temperatures averaging near 26 ° C (78 d F) and April the warmest with temperatures from 33 to 36°C (91 to 96 ° F). The climate is one of the wettest in the world. Most places receive more than 1,525 mm of rain a year, and areas near the hills receive 5,080 mm). Most rains occur during the monsoon (June-September) and little in winter (November-February). Cyclones, floods, and similar disasters create unrecoverable damages in people's lives. However, if people were informed about an upcoming cyclone, flood, or similar disaster, they could take required actions. To prevent or diminish the level of severity of such incidents, we decided to help people in Bangladesh especially in the Cox's Bazar district.

Figure 1.: Coastal Map Of Bangladesh

Cox's Bazar is located 150 km (93 mil) south of the divisional headquarter city of Chittagong. The Cox's Bazar town has an area of 6.85 km^2 (2.64 sq. mi) and is bounded by Bakkhali River on the north and East, Bay of Bengal in the West, and Jhilwanj Union in the south.

As Cox's Bazar is an important area from this standpoint, it has also been subjected to different historical events. One of those issues is so called Rohingya crisis, which significantly affected the area. In the Rohingya crisis hundreds of thousands of terrified Rohingya refugees began flooding onto the beaches and paddy fields of southern Bangladesh in August 2017. By the end of August 2021, Bangladesh was hosting more than 890,000 Rohingya refugees in the Cox's Bazar District, around half of whom were children. we think that it would be very important for the people of this area to be well-informed about the weather conditions. This way thousands of people can save their assets, belongings and their lives as well.

[5]

Figure 2.*: Rohingya Refugee Camp*

4. REQUIREMENTS AND METHODOLOGY

4.1 User -friendly Application

This web application is created for the people who are living in the coastal part of Bangladesh. The requirement is to generate alert for next five days for them and give them the update of rainfall, wind gust , heat wave and cold wave. It is important that this web application is user friendly because the data in this application can have impact on the daily-to-daily lives of people living there.

The main page of the web application shows ten days of temperature for the Cox's Bazar district and its eight sub districts. The next part of web application shows the following updates of rainfall, wind gust, heat wave and cold wave in their respective pages. When the user selects the district, the user can see the updates of the forecast for the next five days in the web application. Also, the weather forecast updates will show the user that if the current updated forecast is normal, moderate, heavy or extreme. Angular has been used to create this single page web application for heavy use.

METHODOLOGY

4.2 Preliminary Design

First we fetch data from the local server to mysql database. We can access this data from the backend application. In the backend application using Json we can create three layers using API.

Those three layers are

- Dissemination
- Frontend Application
- Alert

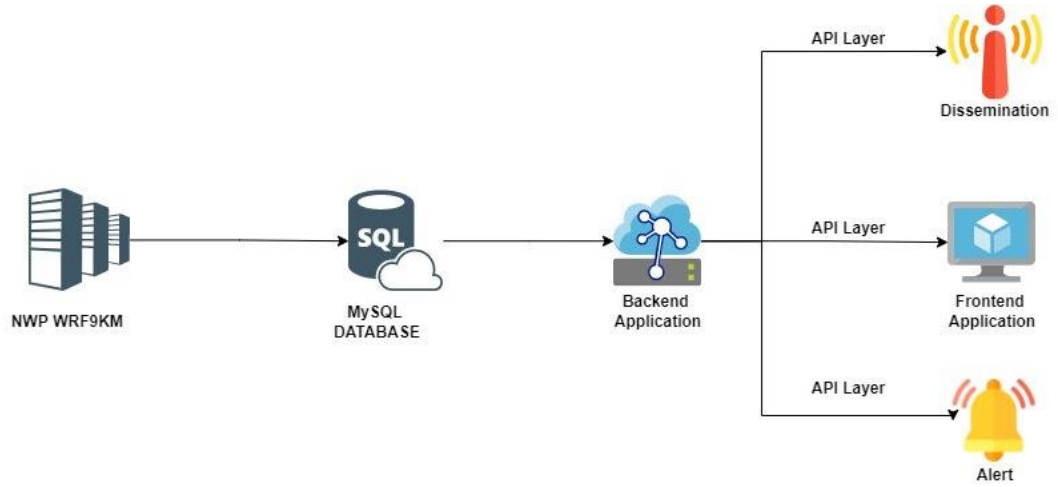

Figure 3.: Web Application Architecture

In the dissemination layer we will be able to send alert to the user where the user will be notified in the user's device. In the second API layer we can see the frontend application where we will create the web application. This web application will show the five days of alert and update it regularly through the API. Lastly, an alert mainly will mainly be in the homepage of application where, if any emergency alert is triggered, it will immediately show.

4.3 Analysis

Before implementing the project, we have made a list of requirements and checked possible needed technology stacks, in order to provide a maintainable, portable and robust application. After an analysis, we have come into the result that we needed the following technologies suitable and sufficient for our needs

- HTML, CSS, Bootstrap
- Angular
- Highcharts
- OpenStreetMap, Leaflet

The technologies used in the application are described in this section.

HTML(Hypertext Markup Language) allows programmers to create and maintain web pages using different HTML Elements, such as paragraphs, links, tags, images, attributes.

CSS (Cascading Style Sheets) is a styling language for depicting the Web pages, allowing us to change colours, layouts, and fonts. It allows us to adapt the presentation to different device types, such as tablets, small phones and modern laptops.

Bootstrap is a powerful CSS helper tool, a well-built suite of HTML, CSS and JavaScript for creating and building web components. It basically makes the creation of commonly used web components for the webpages super easy by providing fully functional components.

Angular is an open-source JavaScript framework created by Google and is written in Typescript. It helps developers to create single-page web applications. Compared to other JavaScript libraries, such as React, Angular is a framework meaning it has its own mechanism to handle logging, running and production ready components.

Highcharts is a library that comes with all the tools needed to create reliable and secure data visualisations. Built on JavaScript and TypeScript, all our charting components are created by this library and work with any back-end database or server stack.

OpenStreetMap is a free worldwide map. OSM is a database product and is edited by several open-source software projects. There are also open-source projects and proprietary products that allow viewing and routing on maps created with the data.

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps, weighing just about 40 KB of JS, it has all the mapping features for most developers according to the official website. It is designed with simplicity, performance and usability in mind.

4.4 Design

To create an online weather forecast tool, the idea was as follows: we could show the following info for five consecutive days:

- Rainfall Alert
- Wind gust Alert
- Heat wave Alert
- Cold wave Alert

We could, in particular, use the following subdistricts of the area to inform the people:

- Upazila
- Chakira
- Sadar
- Kutubdia
- Maheskhali
- Pekua
- Ramu
- Teknaf
- Ukhia

In the following part, the appearance of the application is presented.

On the main page of our application, we provide very valuable information about different types of alerts. On the bottom left, the heat information is given about a selected sub-district. A sub-district can be changed.

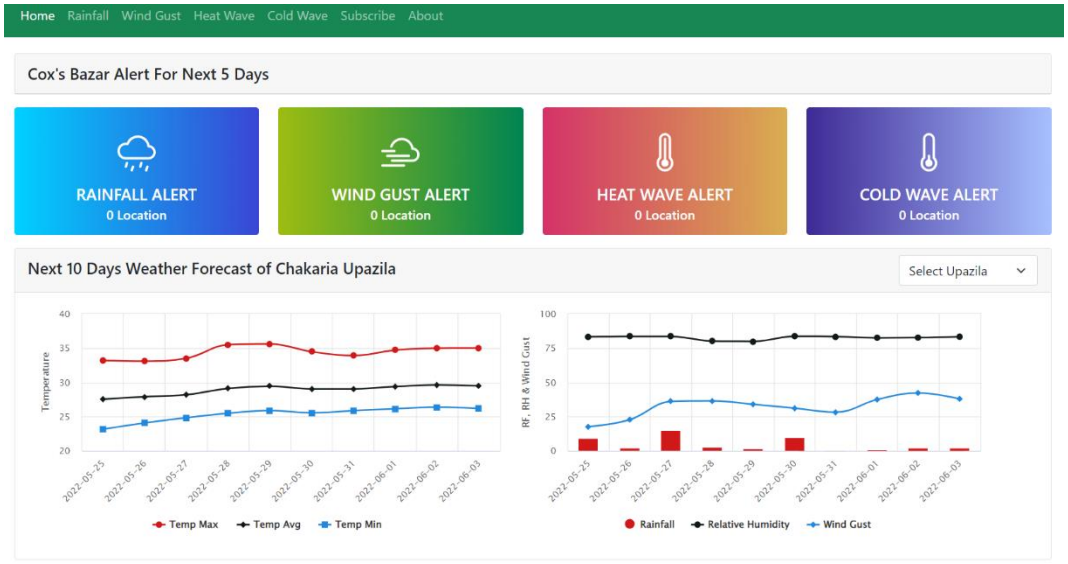

Figure 4.: Main page of web application

On the top right, there will be a dropdown menu where different sub-districts are shown. The different types of alerts, with rainfall, relative humidity and wind gust are shown. Based on the 5-day data information, if any of the parameters are close to thresholds, the alert components on the top are updated accordingly.

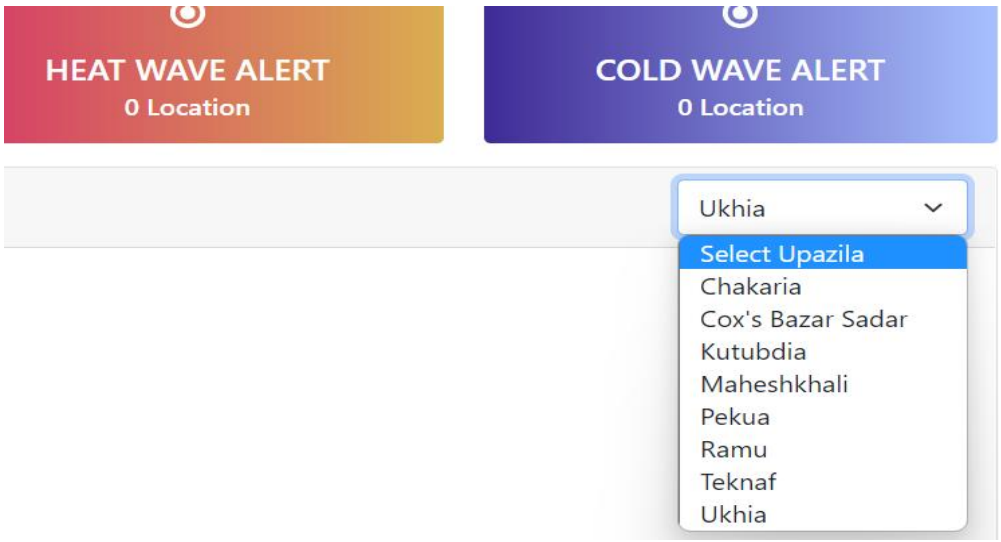

Figure 5.: Dropdown Menu

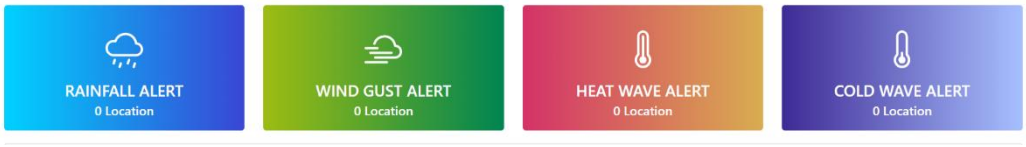

Figure 6.: Alert Option Names

Where the rainfall alert will be shown, initially that area will be marked with different colours based on the rain forecast. Moving forward to Wind Gust alert, Heat Wave alert and Cold Wave alert we will also show the area marked with different colours based on the forecast.

5. IMPLEMENTATION

In this section, we will give an insight into the implementation of our weather forecast web application. We will start by explaining of the source code to show how the actual implementation handles the complex logic in order to create the programming logic behind our application.

5.1 Structure

In this section, we will demonstrate our programming stack. In the previous section, we mentioned that we used a popular Google based JavaScript library, Angular. Angular applications are built by Angular components. Therefore, we will first start with the working logic of Angular Framework.

Angular follows component-based architecture, in component-based architecture, a large application is broken into functional and logical components. These components then are reusable, hence can be used in any other part of the application. From the standpoint of modularity, it is no different from any other object-oriented programming language logic. Having reusable and independent components, we can easily test our components. This way, we can combine as many components as we want and also call them anywhere we want. There are seven building blocks which create an angular application.

1. Modules
2. Components
3. Services
4. Templates
5. Metadata
6. Databinding
7. Directives
8. Dependency Injection.

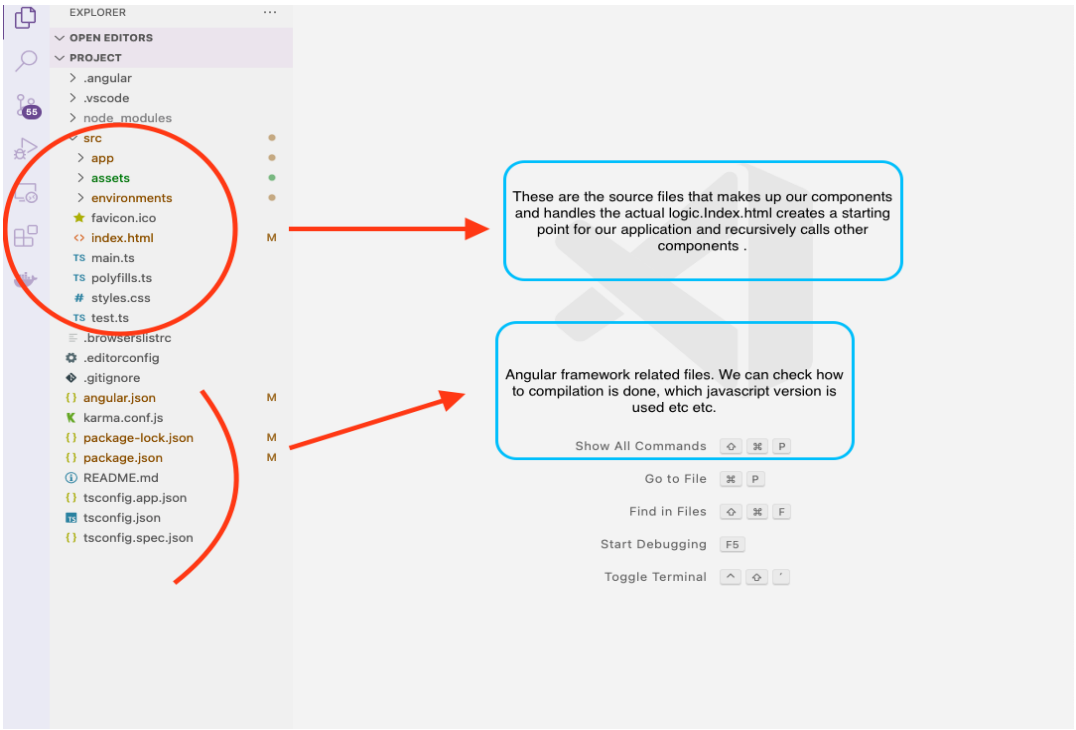

Figure 7 : Structure of the project

Above, we can see the structure of our Angular project. Let us give an overview about the structure. As can be seen from the image above, we have multiple folders.

The **Node_Modules** folder is responsible for having the dependency modules, so that each JavaScript application has dependencies on other libraries, modules or frameworks. In this case, these modules have to be downloaded from somewhere and stored in the application so that they can be used later on. Here, node_modules has the responsibility of that. It holds the downloaded modules from NPM.

The **SRC** folder, as the name stands for, represents the source folder as in any other programming language/framework. Here, we store our application related files, such as classes, services, HTML pages, CSS files and so on.

The **.gitignore** file is a default file which is used by Git to understand which files should be ignored to store the application on the git server.

5.2 Demonstration & Code Architecture

In the following part, we will start by showing how to run our application and then show each UI part and show the underlying implementation logic behind that. Each UI part will be explained in combination with the source code details. This way, it will be clear how each component works with the corresponding data handling.

Let us start by running our project and show how it is presented to the end user. Angular framework provides all the means to our application to run it.

We can use the “ng serve” command to run our application from the terminal.
Let us do it and see the application.

```
vendor.js              | vendor       |   2.77 MB |
polyfills.js           | polyfills    | 294.85 kB |
styles.css, styles.js  | styles       | 187.57 kB |
main.js                | main         | 105.47 kB |
runtime.js             | runtime      |   6.52 kB |

                       | Initial Total |   3.35 MB

Build at: 2022-05-08T17:25:29.756Z - Hash: a625d22fcf0a4cc6 - Time: 3989ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

Figure 8.: Build log after starting server

As can be seen above, in the terminal we have run the “ng serve” command and the application files are compiled. This compilation logic and rules are applied from angular.json and tsconfig.json files. We can see that our application is served at the port number of 4200. Now, if we open our browser and the stated localhost link with the port number of 4200, we will see that our application is served there. Below, we can see how the applications main page is served at the specified URL.

In the following parts, we will continue with the demonstration of each page, as well as their working code logic.

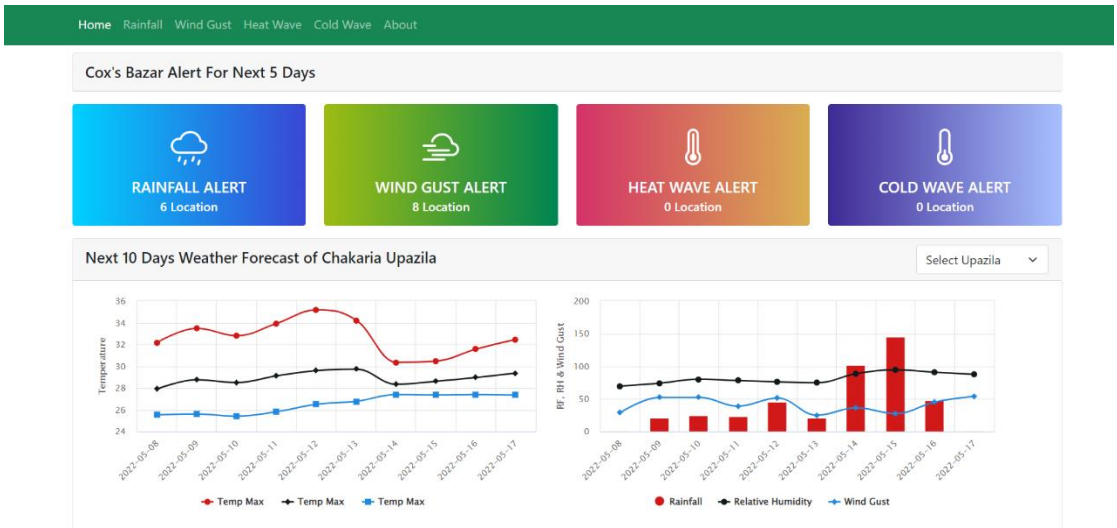

Figure 9.: Homepage of the web application

Now let us take a look at the app-navbar.html component which is the basic menu helping us to navigate between different menu items.

```
<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-success">
  <div class="container">
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLinkActive="active"
routerLink="home">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLinkActive="active"
routerLink="rainfall">Rainfall</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLinkActive="active"
routerLink="windgust">Wind Gust</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLinkActive="active"
routerLink="heatwave">Heat Wave</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" aria-current="page" routerLinkActive="active"
routerLink="coldwave">Cold Wave</a>
```

```
      </li>
      <li class="nav-item">
        <a class="nav-link" aria-current="page" routerLinkActive="active"
routerLink="about">About</a>
      </li>
    </ul>
   </div>
  </div>
</nav>
```

Code Snippet 1.: Router link for every component

When we click on the app-navbar component, on the left-hand side of the IDE, we can see its component and corresponding files are opened. To see how it looks, let us check the html file. In code snippet 1 we can see the HTML file content. Different routes are defined with the values of Home, Rainfall, Wind Gust, Heat Wave, Cold Wave and About. These are simple HTML tags. However, what is important to know is that the Angular framework provides means to switch between different routes. This route switching is handled by the “routerLink” directive. As an example, we marked one of the directives on top of the screen. As this being the case, Angular framework should know how the value of routerLink should open the correct page. To solve this problem angular framework allows us to define different paths and their corresponding components in the app routing module file. Let us show the contents of this file.

```
const routes: Routes = [{
  path: "",
  pathMatch: "full",
  redirectTo: "home"
 }, {
  path: "home",
  component: HomeComponent
```

```
}, {
  path: "rainfall",
  component: RainfallComponent
}, {
  path: "windgust",
  component: WindgustComponent
}, {
  path: "heatwave",
  component: HeatwaveComponent
}, {
  path: "coldwave",
  component: ColdwaveComponent
}, {
  path: "about",
  component: AboutComponent
}
];

@NgModule({
 imports: [RouterModule.forRoot(routes)],
 exports: [RouterModule]
})
export class AppRoutingModule { }
```

Code Snippet 2.: Paths for components

In Code snippet 2, we can see the different route values within the routes Array. Each element has a path value which then is used in the app-navbar routerLink value. The corresponding component is then called to handle the rendering operation. Note that Angular is a single page application, so when a new page is

rendered the browser does not refresh the page itself, rather the page is repainted with the component replacement operation.

We have successfully shown the navigation operations. Now, we can start to show some of the page and their working logic.

If we open the Home page component, we will see the HTML code shown in Code snippet 3. What the code does is that in the first lines it renders the elements for generalised Alerts.

It does this by using the home.component.ts file. As can be seen from the TS file snippet below, alert count variable is initially set to 0, and is updated based on the configurations for each selection.

```
export class HomeComponent implements OnInit, AfterViewInit {

 Highcharts: typeof Highcharts = Highcharts;
 updateFlag = false;
 chartOptions: Highcharts.Options = {
  series: [ { type: 'spline', data: [] }, { type: 'spline', data: [] }, { type: 'spline', data: []
} ]
 };
 chartOptionsRain: Highcharts.Options = {
  series: [ { type: 'column', data: [] }, { type: 'line', data: [] }, { type: 'line', data: [] } ]
 };

 alert_count = {
  alert_rain: 0,
  alert_heat: 0,
  alert_cold: 0,
  alert_wind: 0
```

```
};
upazila_name = "";
forecast_data:any;
constructor(private http:HttpClient, private service: GenerelService) { }
```

Code Snippet 3.: Parameters for alert

We can see that initially all alert values are set to 0. Afterwards, the ngOnInit method which then recursively calls the getAlert() method updates the values.

```
async getAlert() {
  await this.service.getAlertCount().subscribe(data=>{
    this.alert_count = data;
    console.log(this.alert_count);
  });
}
```

Code Snippets 4 : getAlert function

In code snippet 4 we can see that the service is used to get alert counts. Every time an event is emitted from the service, the data is subscribed so we update the value of alert_count variable and then update the value. When the data is updated, we are able to print it in the HTML page by using the string interpolation feature of the Angular framework as follows.

```
<div class="col-lg-3">
    <div class="card rain_bg">
      <div class="card-body text-center">
        <svg xmlns="http://www.w3.org/2000/svg" width="48" height="48"
fill="currentColor" class="bi bi-cloud-drizzle my-3" viewBox="0 0 16 16">
          <path d="M4.158 12.025a.5.5 0 0 1 .316.633l-.5 1.5a.5.5 0 0 1-.948-
.316l.5-1.5a.5.5 0 0 1 .632-.317zm6 0a.5.5 0 0 1 .316.633l-.5 1.5a.5.5 0 0 1-.948-
```

```
.316l.5-1.5a.5.5 0 0 1 .632-.317zm-3.5 1.5a.5.5 0 0 1 .316.633l-.5 1.5a.5.5 0 0 1-
.948-.316l.5-1.5a.5.5 0 0 1 .632-.317zm6 0a.5.5 0 0 1 .316.633l-.5 1.5a.5.5 0 1 1-
.948-.316l.5-1.5a.5.5 0 0 1 .632-.317zm.747-8.498a5.001 5.001 0 0 0-9.499-
1.004A3.5 3.5 0 1 0 3.5 11H13a3 3 0 0 0 .405-5.973zM8.5 2a4 4 0 0 1 3.976
3.555.5.5 0 0 0 .5.445H13a2 2 0 0 1 0 4H3.5a2.5 2.5 0 1 1 .605-4.926.5.5 0 0 0
.596-.329A4.002 4.002 0 0 1 8.5 2z"/>
          </svg>
          <h5 class="card-title">RAINFALL ALERT</h5>
          <h6 class="card-subtitle">{{alert_count.alert_rain}} Location</h6>
        </div>
      </div>
    </div>
```

Code Snippets 5 : Handling of graph

The logic of the warning counts is shown by far. Now we can show how the graph implementation is handled. Earlier we mentioned that we use HighChart for our project.

In Figure 10, we marked three different sections. These three markers explain the logic for the important concept. Number 1 is the function which is called each time we select a new value from the dropdown list. If we hover over the onOptionsSelected() method, we can be able to navigate in it.

Figure 10.: Dropdown list for sub districts

When we call the method then at code snippet 6 we can see that the getForecast() method is then called.

```
onOptionsSelected(value:string){
  console.log("the selected value is " + value);
  this.getForecast(value);
}
```

Code Snippet 6 . forecast calling

The getForecast method then subscribes to the service and retrieves the data. Each time a new event is emitted through this method, data is updated, and this update is reflected through the genHighChartOptions method.

```
async getForecast(upazila_id:any) {
  await this.service.getLocationForecast(upazila_id).subscribe(data=>{
    this.upazila_name = data.result.location_info.name;
    this.forecast_data = data.result;
```

```
    this.genHighchartsOptions(data.result);


  });
}
```

Code Snippet 7.: getForecast function

Number 2 shows the possible input values, districts in our case, to update the function. Finally, in number 3 where the chart is marked, we provide two different maps and through the help of the HighChart the values are updated accordingly.

Our second menu item is Rainfall. In the rainfall section, a typical image view looks as follows.

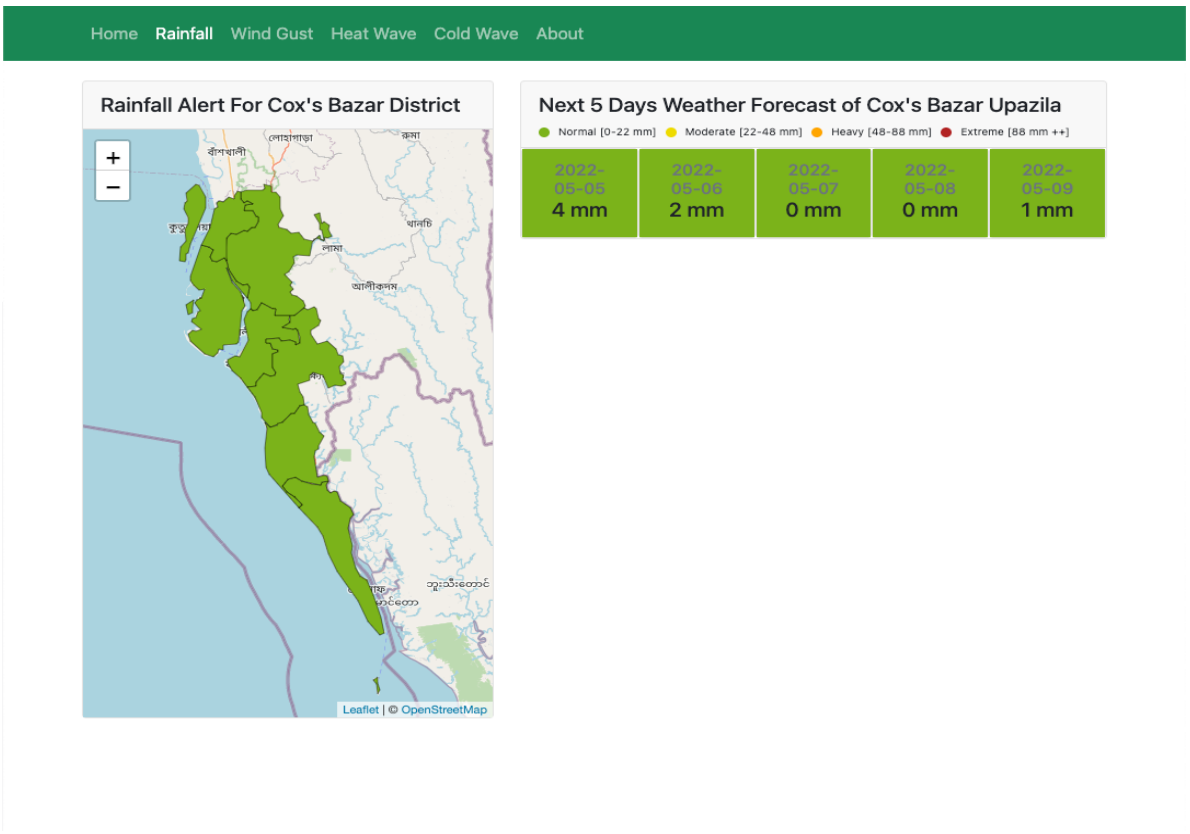

Figure 11.: UI for rainfall

We can see from the above image that the selected district is visually highlighted on the left-hand side with the help of our map service. On the right-hand side, the next five days precipitation is written in a table. We can see that on 05-05-22, there will be 4mm precipitation expectation while the following days are expected to have 2mm, 0mm, 0mm, 1mm, respectively.

In the below code snippet, we can see the HTML content of the Rainfall page to render such a page. Above the file, we have the map div which then helps us render the page. Later, line numbers between 13-33 in the file helps us to create the table headers. And the important part is between line number 36,and 43 in the file where the alert values are iterated and the item element's properties are used to display different values of the days.

```
<div class="row mb-5">
  <div class="col-lg-5 mt-3">
    <div class="card text-start">
      <div class="card-header">
        <h5 class="mb-0 py-1 card-title">Rainfall Alert For Cox's Bazar
District</h5>
      </div>
      <div class="card-body p-0">
        <div id="map"></div>
      </div>
    </div>
  </div>

  <div class="col-lg-7 mt-3">
    <div class="card text-start">
      <div class="card-header">
        <h5 class="mb-0 py-1 card-title">Next 5 Days Weather Forecast of
{{selected_upazila}} Upazila</h5>
```

```
          <div class="d-flex justify-content-start mt-1" style="display: block;">
            <div class="d-flex align-items-center">
              <div style="width: 10px; height: 10px; border-radius: 50%;
background-color: #7BB31A;"></div>
              <p class="py-0 my-0 mx-2" style="font-size: 10px;">Normal [0-22
mm]</p>
            </div>
            <div class="d-flex align-items-center ml-1">
              <div style="width: 10px; height: 10px; border-radius: 50%;
background-color: #EEDB00;"></div>
              <p class="py-0 my-0 mx-2" style="font-size: 10px;">Moderate [22-
48 mm]</p>
            </div>
            <div class="d-flex align-items-center ml-1">
              <div style="width: 10px; height: 10px; border-radius: 50%;
background-color: #FFA500;"></div>
              <p class="py-0 my-0 mx-2" style="font-size: 10px;">Heavy [48-88
mm]</p>
            </div>
            <div class="d-flex align-items-center ml-1">
              <div style="width: 10px; height: 10px; border-radius: 50%;
background-color: #B22222;"></div>
              <p class="py-0 my-0 mx-2" style="font-size: 10px;">Extreme [88
mm ++]</p>
            </div>
          </div>
        </div>
        <div class="card-body p-0 d-flex">
          <div class="card flex-fill" *ngFor="let item of alert_value">
            <div class="card-body text-center alert_{{item.alert}}">
```

```
                <h6 class="card-subtitle text-
muted">{{item.step_start.substring(0,10)}}</h6>
                <h5 class="card-title mb-0">{{item.result}} {{item.unit}}</h5>
              </div>
            </div>
          </div>
        </div>

</div>
```

Code Snippet 8. Fetching value for alert

The initMap method is used to initialise the map based on the input provided by the user. Afterwards, the fetchAlertUpazila method is called to retrieve the data.

```
intiMap():void {
  this.map = L.map('map', {
    center: [ 21.30, 92.10 ],
    zoom: 9
  });

  const tiles = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    maxZoom: 18,
    minZoom: 3,
    attribution: '&copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
  });

  tiles.addTo(this.map);
```

```
  this.fetchAlertUpazila(2022, 1);
}
```

Code Snippets 9. Integrating leaflet for rainfall

We have used this similar method in other navigation menus which are wind gust, heat wave and cold wave.

6. CONCLUSIONS

Climate change is becoming a larger issue day by day. The whole point of developing this web application was to make life easier for people who are living in the coastal part of Bangladesh. The Bangladesh Government took this initiative, and it is the first web application in this down scale that has been created to generate an alert system for this region. This application currently gives five days of upcoming alert for rainfall, wind gust, heat wave and cold wave. To summarize, this web application can give hope to local people who become helpless in front of the disasters. A rainfall alert can help farmers to save their livestock in crucial time, a wind gust alert can be helpful for the fisherman, a heat wave or cold wave alert can be vital for older people because of their health situation. So, the objective of this thesis was successfully implemented and goals were achieved. But it is possible to take this web application to the next phase targeting the whole country and by analysing the needs of people in different areas of Bangladesh. For example, Bangladesh is also an agricultural country. During the summer life becomes more miserable for the people who are living in the northern part of Bangladesh because of heat waves. If we can develop this web application can be developed for the entire country, it is possible to create a subscription alert for people to make their daily-to-daily life easier. To get the subscription alert it is necessary to have the email address or contact number.. To take the system to the next level, we have to understand that Bangladesh is still on their way of becoming a developing country. The world is becoming advanced through the technology day by day and so is Bangladesh. At the moment not all citizens in Bangladesh have access to modern technology but hopefully in the future they have and thus have access to this application, as well.

REFERENCES

1. World Meteorological Organisation. 2021. Updated 30-year reference period reflect climate change. Accessed xx.xx.2022 https://public.wmo.int/en/media/news/updated-30-year-reference-period-reflects-changing-climate (05.05.21)
2. Haines, Andy, and Jonathan A. Patz. Year? Health effects of climate change. *Jama* 291.1 (2004): 99-103.
3. Compant, Stéphane, Marcel GA Van Der Heijden, and Angela Sessitsch. Year? Climate change effects on beneficial plant–microorganism interactions. FEMS microbiology ecology 73.2 (2010): 197-214.
4. Hardy, John T. 2003. Climate change: causes, effects, and solutions. Edition, printing place?John Wiley & Sons.
5. https://www.hrw.org/report/2018/08/05/bangladesh-not-my-country/plight-rohingya-refugees-myanmar *"Bangladesh is not my country":* (05.08.2018)