Assignment Self Evaluation

Yannan Zhang

2304473

Software Engineering

WEBSITE URL: WEATHER STATION

URL WITH DNS: WEATHER DASHBOARD

WHAT DOES IT HAVE

FOUR VIEWS

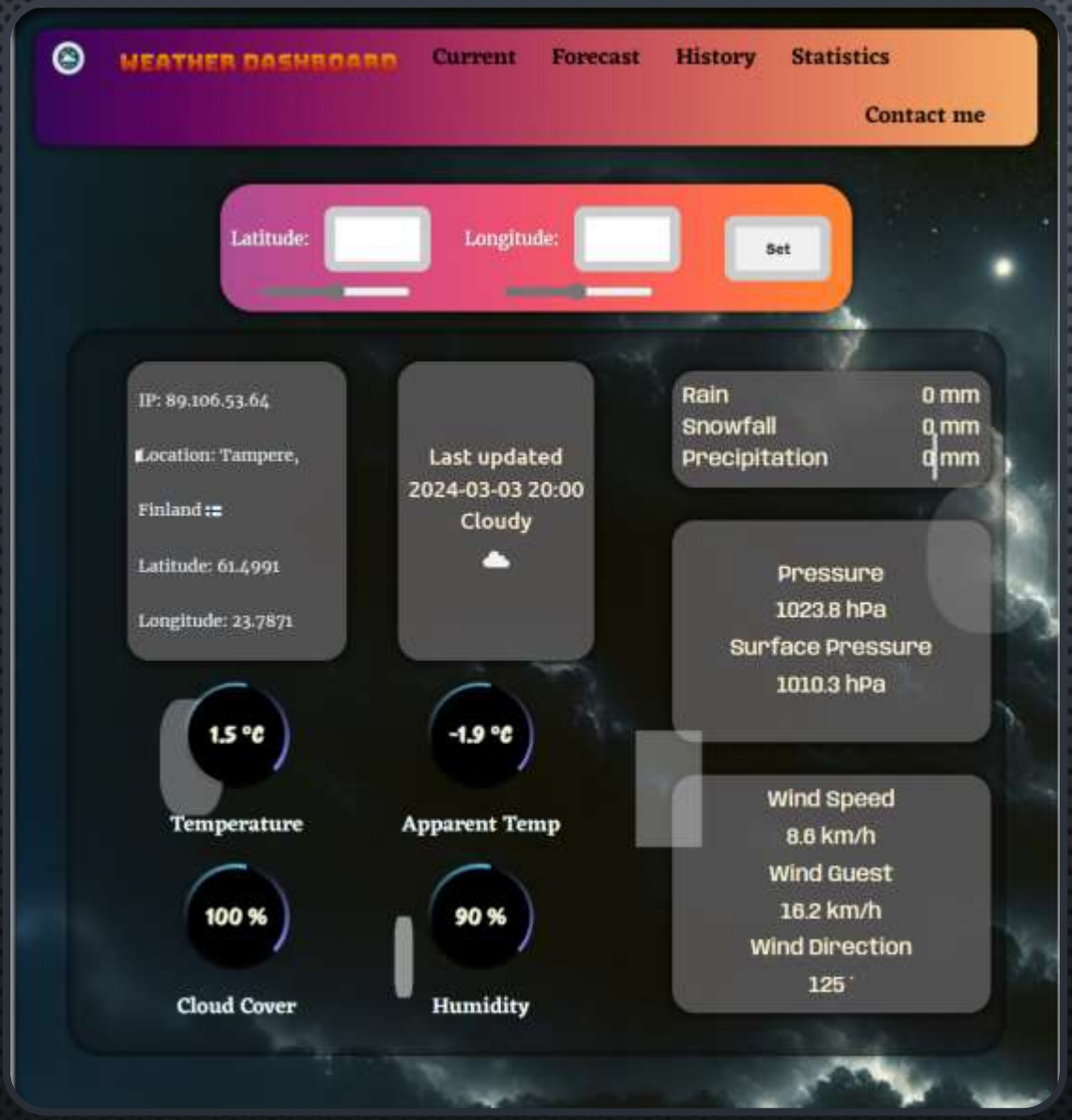
BACKGROUND: DYNAMIC BACKGROUND IMAGE, ACCRODING TO CURRENT WEATHER

IN CURRENT: IP/ LOCATION/ LATITUDE/ LONGITUDE/ CURRENT WEATHER/ PRECIPITATION/ AIR PREESURE/ WIND DETAILS

IN FORECAST: 7 DAYS WEATHER STATUS/ 7 DAYS HOURLY CHART (TEMPERATURE/ CLOUD COVER/ WIND SPEED/ SUNSHINE DURATION/ PRECIPITATION POSSIBILITY)

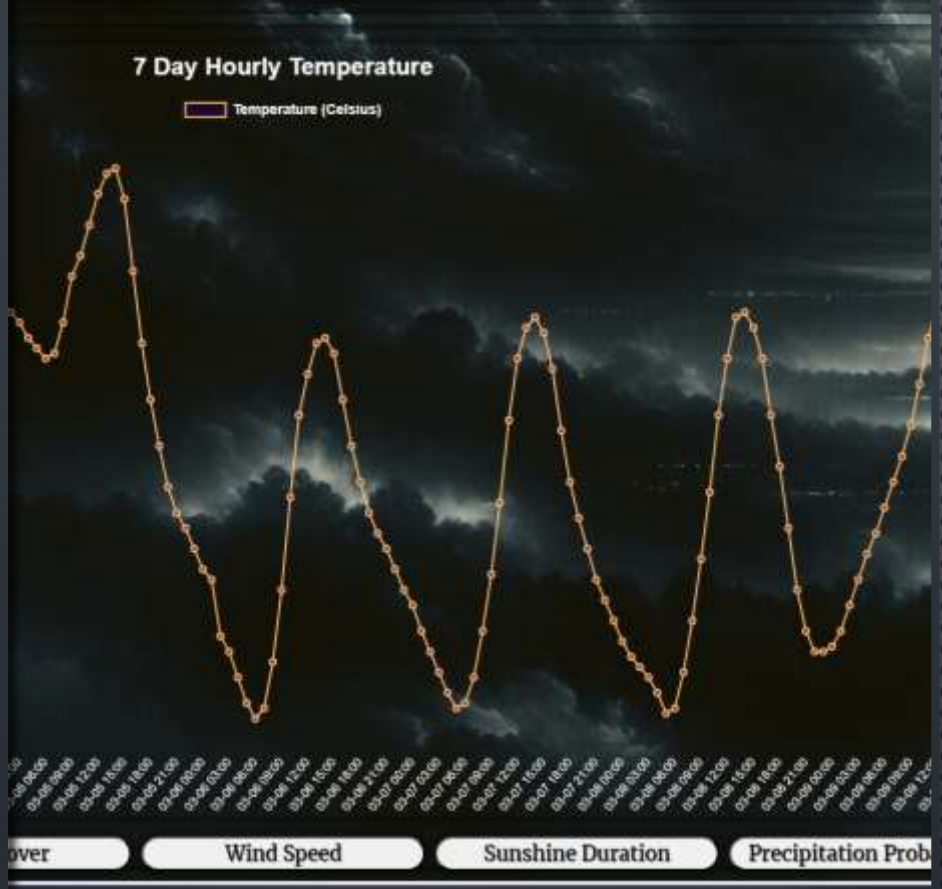
IN HISTORY: FROM 1 TO 90 DAYS WEATHER STATUS SEARCH (20 SEARCH CONDITIONS)

IN STATISTICS: 7 MATH FUNCTIONS FOR CALCULATING NEXT 7 DAYS, AND 7 DAYS TOTAL



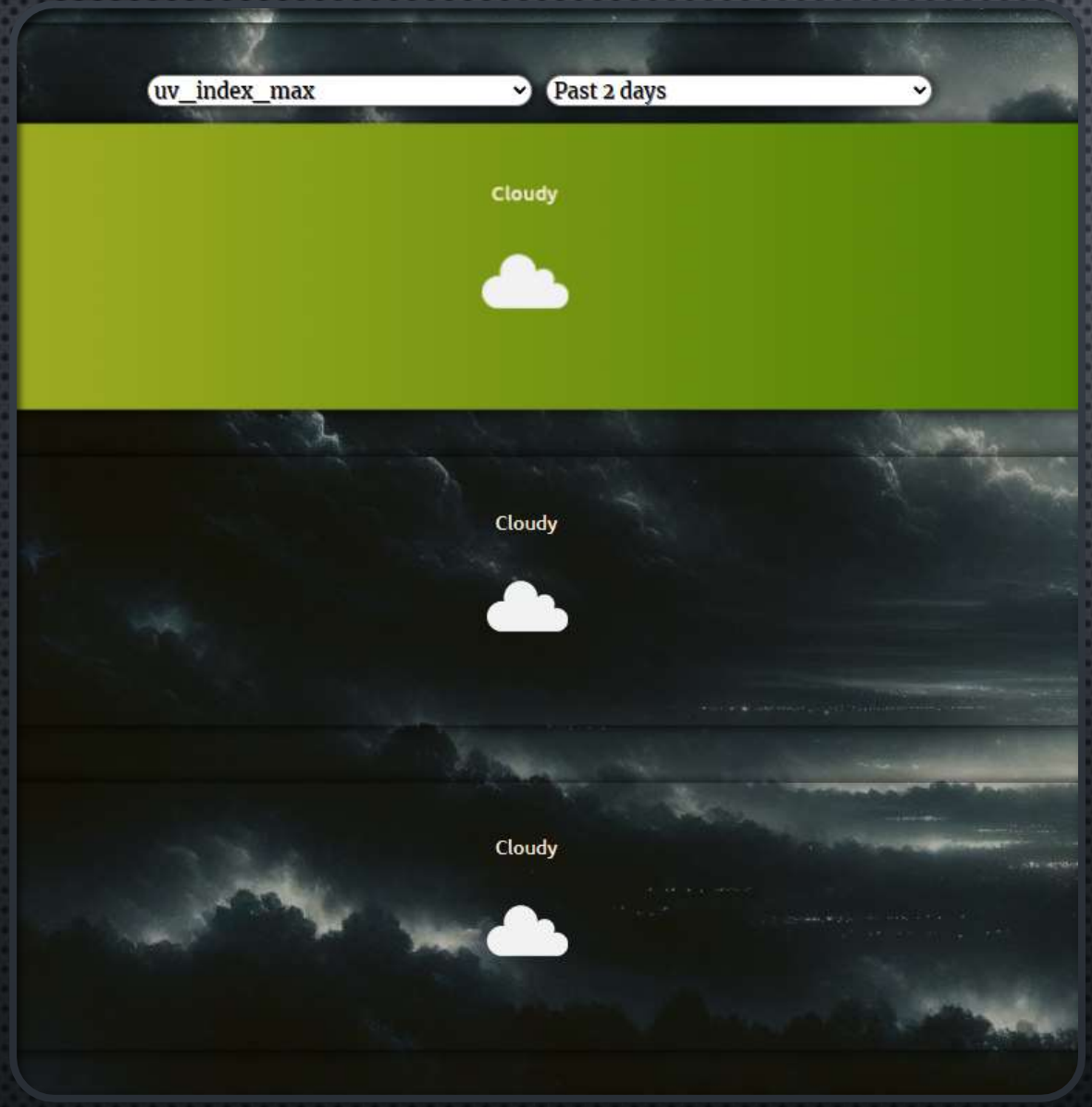
PRODUCT DISPLAY

PAGE CURRENT



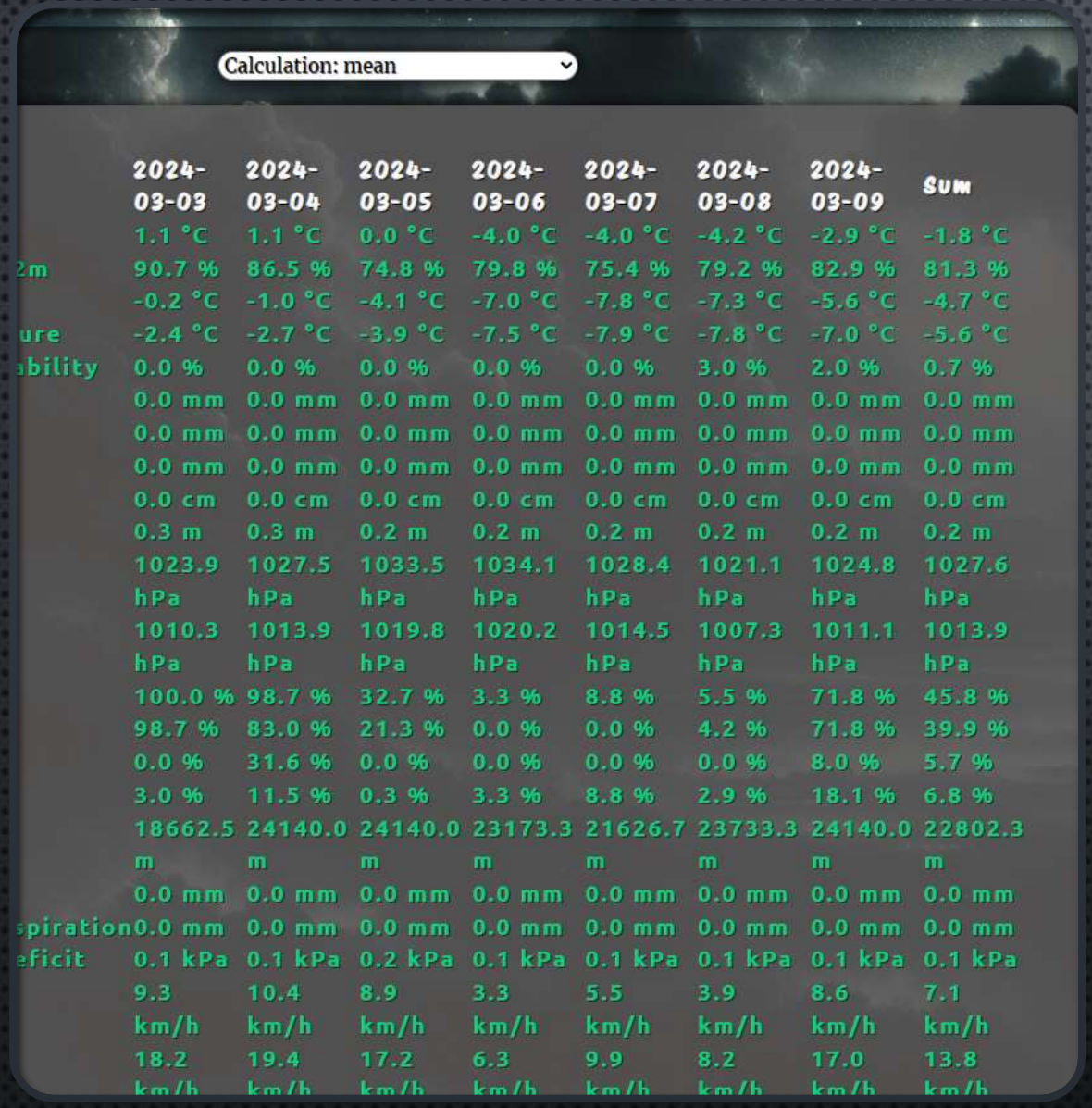
PRODUCT DISPLAY

PAGE FORECAST



PRODUCT DISPLAY

PAGE HISTORY



PRODUCT DISPLAY

PAGE STATISTICS

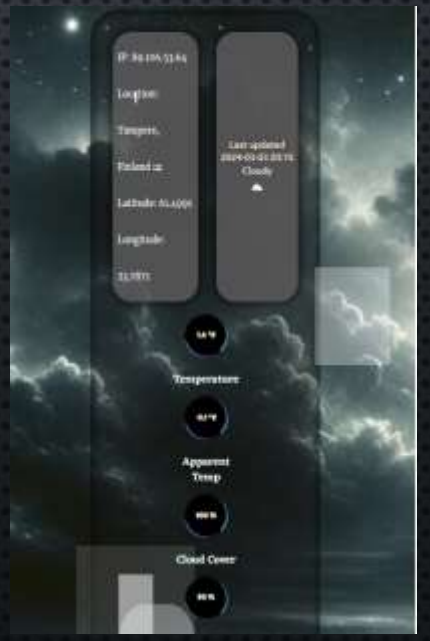
WHAT WENT WELL

- 1. HTML:
- USING BOOTSTARP NAV(AND FOOTER) STRUCTURE
- CLEAR LAYOUT, AND ENOUGH COMMENT

- 2. CSS:
- RESPONSIVE CSS DESIGN
- VIEWPORT UNIT IMPLEMENT
- RESIZE CHART AND IMAGE (BACKGROUND)

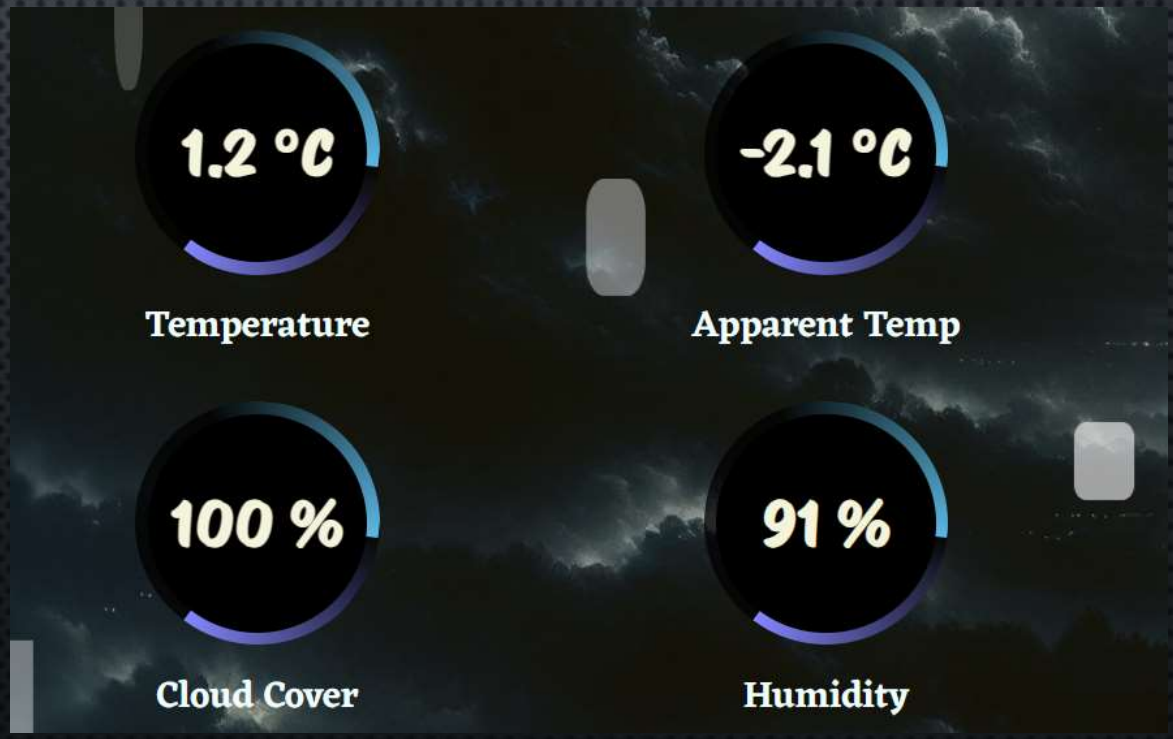
```
<body>
    <nav id=navbar class="navbar navbar-inverse navbar-expand-md nav-round">···
    </nav>
    <!-- form to change coords -->
    <form name="coords" action="submit" method="post" id="coords">···
    </form>
    <div id="container"></div>
    <!-- statistics JS -->
    <script src="../asset/js/statistics.min.js" type="text/javascript"></script>
    <!-- jQuery -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <!-- Bootstrap JavaScript -->
    <script src="https://cdn.bootcdn.net/ajax/libs/twitter-bootstrap/3.4.1/js/bootstrap.min.js"
        integrity="sha384-aJ21OjlMXNL5UyIl/XNwTMqvzeRMZH2w8c5cRVpzpU8Y5bApTppSuUkhZXN0VxHd"
        crossorigin="anonymous"></script>
    <!-- chart JS -->
    <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.1/dist/chart.umd.min.js"></script>
    <!-- moment JS -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.30.1/moment.min.js"></script>
    <script>
        moment().format();
    </script>
    <!-- my JS -->
    <script defer src="./script.js"></script>
</body>
```

```
@media screen and (max-width: 960px) {···
}

@media screen and (max-width: 500px) {···
}

/* list animation */
.squareList-item,
.circleList-item {···
}

.squareList-item {···
}

.squareList li:nth-child(2) {···
}

.squareList li:nth-child(3) {···
}

.squareList li:nth-child(4) {···
}

.squareList li:nth-child(5) {···
}
```



WHAT WENT WELL

- 3. CSS:
- SMOOTH TRANSFORM FOR HOVER AND FOCUS,
- ANIMATION FOR ::BEFORE AND ::AFTER
- DYNAMIC SCREEN ANIMATION



WHAT WENT WELL

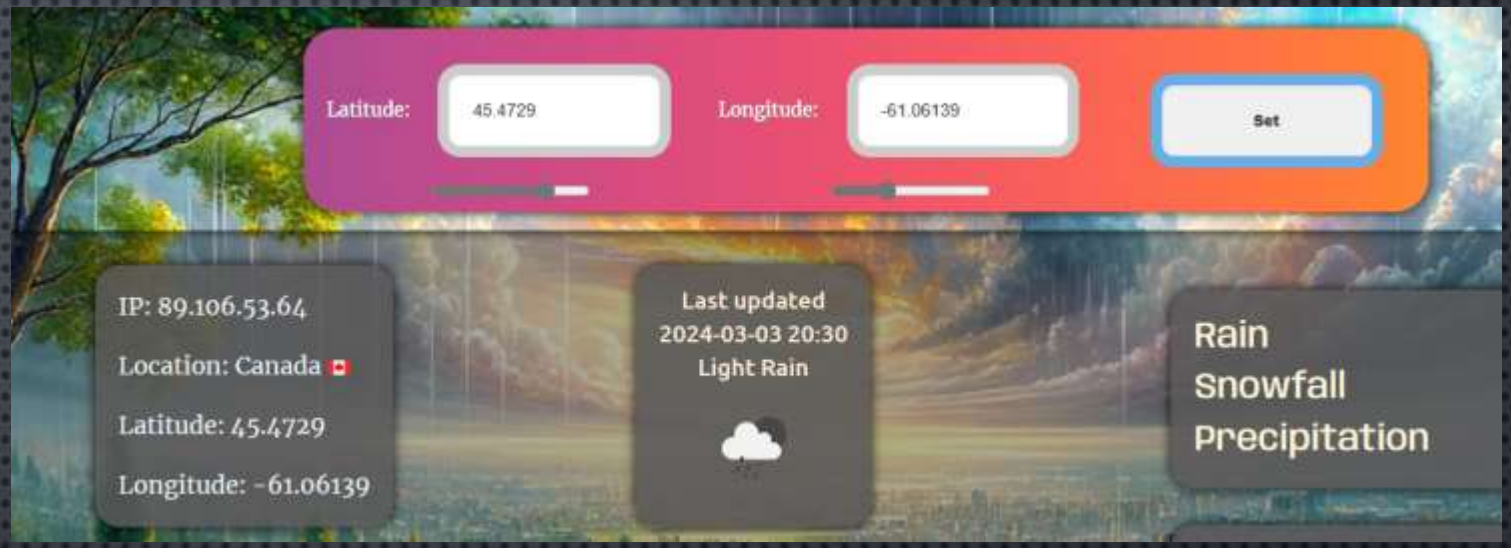
- 4.JAVASCRIPT:
- CLEAR STRUCTURE AND READABLE NAME OF FUNCTION AND VARIABLES
- FRONTEND AUTO REFRESH

```
1155  // update statistics page
1156  const updateStatisticsPage = async (coords) => {···
1158  }
1159
1160  // update background picture
1161  const updateBackgroundPicture = async (coords) => {···
1175  }
1176
1177  // update contact page
1178  const updateContactPage = async () => {···
1203  }
1204
1205  // select weather page
1206  const selectPage = async (coords) => {···
1277  }
1278
1279  //create position select method (user input coordinates)
1280  const updateCoordinate = async (coords) => {···
1309  }
1310
1311  let isFormEventAttached = false
1312  // handle form submit
1313  const handleFormSubmit = () => {···
1331  }
1332
1333  // handle input coordinates
1334  const inputCoords = () => {···
1359  }
1360  // app start
1361  const app = async () => {···
1392  }
1393
1394  app()
1395
1396  // auto refresh
1397  const myRefresh = () => {···
1400  }
1401  setTimeout("myRefresh()", 900000)
```

```
// auto refresh
const myRefresh = () => {
  window.location.reload()
  console.log("Auto refresh")
}
setTimeout("myRefresh()", 900000)
```

WHAT WENT WELL

- 5.JAVASCRIPT:
- FETCH COORDINATES FROM SERVER (WITH A DEFAULT VALUE: TAMPERE)
- CAN SET COORDS WITH USERINPUT




```
//create position select method (user input coordinates)
const updateCoordinate = async (coords) => {
  try {
    console.log(JSON.stringify(coords))
    const response = await fetch(`http://172.16.7.49:5000/api/coords`, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Accept': 'application/json'
      },
      body: JSON.stringify(coords)
    })
    if (response.status === 200) {
      const getResponse = await fetch(`http://172.16.7.49:5000/api/coords`, {
        method: 'GET',
        headers: {···
        }
      })
      const data = await getResponse.json()
      if (getResponse.status === 200) {
        return data
      }
    }
    return null
  } catch (error) {···
  }
}
```

WHAT WENT WELL

- 6.JAVASCRIPT:
- MULTIPLE EVENTLISTENER:
- SELECT PAGE
- BOND TO NAV TAG (WITH USING CSS TRANSFORM)
- ‘CONTACT ME’ CLICK TO GENERATE FOOTER

```
// update contact page
const updateContactPage = async () => {
  try {
    const container = document.getElementById('container')
    let contact = document.getElementById('Contact')
    if (!contact) {
      contact = document.createElement('footer')
      contact.id = 'Contact'
      container.appendChild(contact)
    }
    contact.innerHTML = `
    <div>
      <a href="/">
        <!-- my own icon -->
        <img id="icon" src="../asset/img/icon.png" alt="Brand">
      </a>
      <span>  2024 Yannan Zhang</span>
    </div>
    <ul>
      <li><a href="https://twitter.com/Nancheung_" target="_blank"><i cla
      <li><a href="https://www.instagram.com/nancheung997/" target="_blan
      <li><a href="https://www.facebook.com/profile.php?id=61550541868605
    </ul>`
  } catch (error) {
    console.log(error)
  }
}
```

WHAT DIDN'T GO WELL

- 1.JS:
- NESTED TOO MUCH
- DIDN'T RESUSE IN SINGLE API REQUEST (WASTE MORE FETCH TIME), EACH VIEW HAS MORE THAN ONE FETCH FUNCTION

```
58
59    // update History page
70  > const updateHistory = async (coords) => {···
84    }
85
86    // fetch history weather info
87  > const fetchHistory = async (coords, category = 'temperature_2m_max', past_days = 7) => {···
96    }
97
98    // update history page
99    const updateHistoryPage = async (coords) => {
100     await updateHistory(coords)
101   }
102
```

```
const fetchCurrent = async (coords) => {
  try {
    const apiCurrent = `https://api.open-meteo.com/v1/forecast?lat
    const response = await fetch(apiCurrent)
    const data = await response.json()
    return data
  } catch (error) {···
  }
}
```

WHAT DIDN'T GO WELL

- 2.JS:
- NEED TO GET USER PERMISSION FOR GEO LOCATION

```
// get current address by google map geocode api
const getAddress = async (coords) => {
  try {
    // transform coords promise into object···
    const apiGeo = `https://maps.googleapis.com/maps/api/geocode/json?latlng=$
    const response = await fetch(apiGeo)
    const data = await response.json()
    return data
  } catch (error) {
    return 'Error happened, this brower does not support geolocation.'
  }
}

// get coords
const getCoords = async () => {···
}

// return (navigator.geolocation.getCurrentPosition) -> position
const getLocation = () => {
  return new Promise((resolve, reject) => {
    navigator.geolocation.getCurrentPosition(resolve, (error) => {
      console.error('Geolocation error:', error.message)
      reject(error)
    })
  })
}
```

WHAT DIDN'T GO WELL

2.NODE.JS:

TOO BASIC FUNCTION FOR BACKEND,
NEED PROCESS MORE DATA

WHAT WOULD YOU DIFFERENT NEXT TIME

- 1. HTML: INVOKE MORE COMPONENTS FROM BOOTSTRAP 5
- 2. CSS: MORE ADJUSTMENTS TO THE PROPORTIONS, PREFERABLY SKETCHED FROM THE START (HAVE A DRAW.IO FILE IN PROJECT FOLDER), TO BE MORE RESPONSIVE
- 3. JS: API FETCH REPS REDUCE, MORE DATA VALIDATION, MORE FUNCTIONAL EVENTLISTENER