LMP^2

Lo's Angry Mob
ISO New England

Table of Contents

Team Members and Information

Team Manager

Lo Hazarika	lhazarika@umass.edu

Team members

Ayush Chaudhary	aachaudhary@umass.edu
Alexander Nguy	ajnguy@umass.edu
David Vaykhanskiy	dvaykhanskiy@umass.edu
James Carney	jrcarney@umass.edu
Joshua Shen	jlshen@umass.edu
Maanas Peri	mperi@umass.edu
Savannah Arimento	sarimento@umass.edu
Tai Dang	ttdang@umass.edu
Victor Lee	victorlee@umass.edu

Project Description

Purpose

Design a web application, used by ISO New England clients, that takes a time series of salient energy prices with respect to parameters such as time span, time groupings, node names, and other relevant metrics. Using a specific scenario, produce graphs that compare the scenario data with base case data to check if the scenario data is reasonable with respect to past trends in tandem, to produce data analytics on a set of given measures and time periods.

Users

The main users of our application are Market Analysts, Power Providers, and Finance Companies.

Use Cases

Case 1: Compare a scenario output to the base case to decide if the simulator output looks reasonable.

Case 2: Report on a specific scenario. Once a scenario has been compared to the base case and deemed reasonable, case 1, provide summary statistics for a given measure and time periods.

Constraints

Depending on the querying metric, runtime of the data search may cause the application to function less than optimally since it takes extra time to make the fetch calls. With the large number of individual data nodes, completing queries with multiple parameters will take a sizable amount of time due to the sheer number of comparisons made.

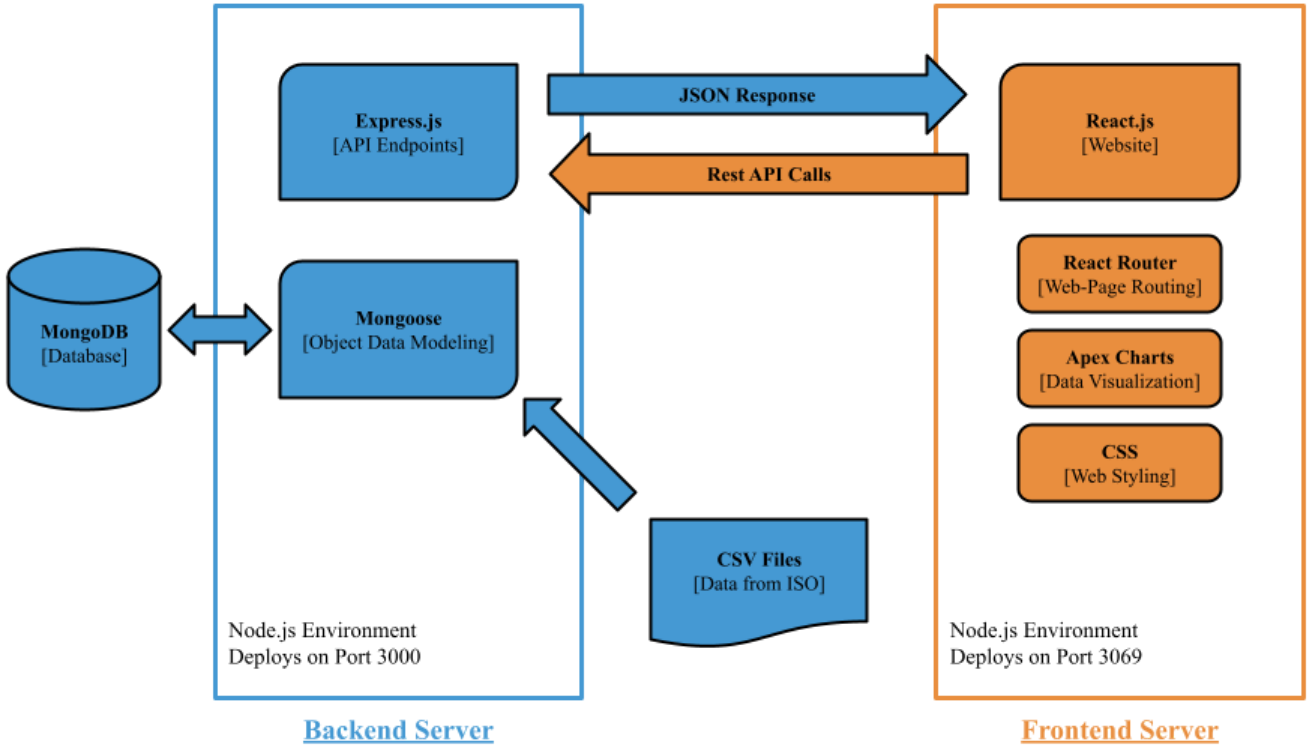
With regard to security, the public nature of the code repository for our web application allows any user to access our product. However, without the specific .env file, they will not be able to access our back-end database, which should be kept private to maintain the integrity of our clients.

Third Party Software Used

Software Title	Description	Purpose
MongoDB	A non-relational document database that provides support for JSON-like storage. MongoDB has a flexible data model that allows for unstructured data with full indexing support.	Where we store, retrieve, and update our relevant data in a secure database.
Mongoose	An unified development framework that allows users to design, build, and deploy applications quickly. Used to establish a connection between MongoDB and Node.js.	Allows us to communicate with the database in a large scalable manner. (i.e. writing a script that generates fake data in masses when we have limited data to work with)
Express.js	A back end web application framework that provides a robust set of features for web applications. Used to build RESTful APIs with Node.js that allow for client and server communication.	Allows for communication of information such as data streams and graphs between our client-side and back-end.
Node.js	A cross-platform environment for executing JavaScript code. Mainly used for server-side development and allows developers to use scripts to create dynamic web page content.	The runtime environment where we deploy our web application to be viewed in the browser.
Node Package Manager (NPM)	A package manager for installing, updating, and uninstalling Node.js packages that are local dependencies of a project.	Used to install the various libraries in packages utilized throughout our web application
React	A JavaScript-based UI development library.	Foundational software of our web application.

React Router	A collection of navigational components that allows for dynamic client-side routing. Allows for single page web applications with navigation without the page refreshing as the user navigates.	Fundamental basis for web application page navigation.
Apex charts	A JavaScript charting library used for creating interactive data visualizations for web pages. Creates embedded analytics using a simple API.	Front-end primary data visualization software.
Cascading Style Sheets (CSS)	A style sheet language used for formatting the layout of web pages. Allows for the separation of webpage content and presentation, which provides flexibility and control over the characteristics. Also allows for uniform formatting over multiple web pages.	Front-end primary styling software.
Bootstrap	An HTML, CSS and JS library that is used to simplify the development of web pages. Using Bootstrap on a web application allows for the use of its color, size, font and layout options to create a uniform appearance across web browsers.	Specific components used throughout the web application such as drop-down menus.
Thunder Client	A fast Rest API client extension on Visual Studio code. Creates requests to retrieve responses quickly and has extensive API request handling.	Backend API testing
JTest	A JavaScript testing framework designed to ensure correctness in any JavaScript codebase.	Frontend Testing

Architectural Diagram of System



The Agile Process

Summary of Our Sprints

Over the course of this project we completed a total of five bi-weekly sprints. Starting on Wednesday, we met as a group to discuss what had been done in the previous sprint and what needed to be completed in the upcoming sprint. From there, the tasks were then delegated to either the front-end or back-end group and were then subsequently assigned to team members. In order to organize and visualize what needed to be done, we used Jira to create an agile board with items in various stages of the scrum process. Along with Jira, we also used a discord server broken up into team specific channels to communicate during each sprint.

At the beginning of the semester, our team was divided into two clearly defined groups, the front-end team and the back-end team. As time went by, around sprint 3, these teams became less clear cut as some people from the back-end hopped onto working on front end tasks. This was mainly due to the back-end work being nearly done, while the front-end still had a lot to complete.

Difficulties with the Scrum Process

- It was hard to self-organize and code at the same time.
- Hard to keep the Jira board and backlog up to date.
- More time was spent on making the weekly presentations than the actual project.
- The demo presentations every week started to feel futile.
- It was hard to remember to check Jira often.

What Worked in the Scrum Process

- Delegating tasks during our weekly meetings.
- The weekly cycles helped us have a sense of our measurable progress.
- Using the Jira board and Discord allowed us to stay organized and continuously communicate throughout the course of the project.
- Allowing remote attendance at meetings allowed for more group members to attend at every meeting.
- Having the Discord organized into general, front-end and back-end channels allowed for members to communicate in an organized manner.

Technical Issues Encountered

- There were issues with getting the basic API integration working.
- It was hard to understand exactly what ISO wanted us to build and there were difficulties in understanding the individual use cases.
- It seemed like the development of the back-end was really slow compared to the front-end. The front-end had to wait for the API implementation.
- Connecting the API to the front-end.
- We were given the sample data so we needed to create more fit with the user cases.
- Issues when trying to build the search bar for the drop down menus.

Testing

Backend

The back-end used Thunder Client to test API calls. They also used unit testing to check the functionality of the back-end by using extended dummy data.

Frontend

The front-end did thorough testing by going through each example. They also tested the compatibility of the application by running it on various browsers on different screen sizes.

Full Stack

Throughout the developmental process, we performed thorough testing by making API calls to the back-end database to fetch data to populate the graphs displayed.

Notes for Installation and Use

In order to install and use our application, make sure to have `npm` installed and have downloaded the `.env` file in the deliverables.

Installation

1. Clone the GitHub repository using the following command

```
% git clone https://github.com/hiimlo/320-ISO-Project-Fall22.git
```

2. Navigate to the root directory (`/320-ISO-Project-Fall22`), execute the following command and paste a copy of the `.env` file

```
% npm i --force
```

Running the Application

1. Navigate to the root directory (`/320-ISO-Project-Fall22`)
2. Execute the following command in terminal

```
% npm start
```

When this command is run, the application starts a script which

a. Navigates to the `/backend` and starts the back-end server
b. Navigates back to the root, starts the front-end server and displays the website

In order to stop the application, simply press `ctrl c` *(or whatever end program key combination your terminal has)*

Link to GitHub Repository

https://github.com/hiimlo/320-ISO-Project-Fall22

Repository Layout