MINI PROJECT

(2020-21)

**MID-TERM REPORT**

E-PRODUCT PRICE TRACKER

***Team Members*:**

**Mayank Kumar Singh (181500378)**

**Priyam Srivastava (181500508)**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



***Supervised by:-***

**Mr. Anand Gupta (*Technical Trainer)***

**Mrs. Ruchi Gupta (*Technical Trainer)***

**GLA UNIVERSITY**

**Mathura – 28140, INDIA**

**Content**

* **Abstract ………………………………………………….. 3**
* **Introduction ……………………………………………… 4**
  1. **General Introduction to the topic**
  2. **Motivation**
  3. **Objective**
* **Problem Definition ……………………………………….. 5**
* **Software Design & Requirement ……………………….. 6**
* **Implementation Detail ………………………………..… 10**
* **Progress till Date & the Remaining work …………….. 12**
* **Screenshots ……………………………………………… 13**

**References ………………………………………………. 16**

**Abstract**

Online websites like Amazon, Flipkart and many more website uses dynamic pricing because of which prices changes periodically depending on market and various reasons.

E Product Price tracker is an online website that tracks prices of an online product and notify the user on the price drop. User can add multiple items and track the prices of the products simultaneously. It saves precious time of user where user does not have to check the website and browse through all the products before buying it. User need not to check daily websites like Amazon, Flipkart our website will do this for the user. We will notify our user through email or user can check directly by logging into our website.

We will use web scrapping technology to get the prices of the product from their respective websites and show them on our website. And HTML for structuring our website and CSS for styling of our website. JavaScript for main functioning of our website. With help of back-end languages like Node.js, ExpressJS, EJS Templating and MongoDB we have used it for storing data and connecting our website to our database.

**Introduction**

**General Introduction:**

E-Product Price tracker is a web application that runs on a MongoDB server, it is built to help in their daily life and reduce their efforts. Central idea of our project is to take input from the user (product URL and price at which they want it) and based on that input we will ping the user whenever there is a price drop and the product is available at their interest price.

**Motivation:**

Many times when we wish to purchase a product online, we get to know that there had been a price drop in the product’s price and we wish if there was a way where we would be notified on the price drop. So we can buy the product at that time. Our website will do this job for our user so the user has not to check on product’s price every day. Our website will notify the user through mail when there is a price drop.

**Objective:**

The Main Objective to create this website is cut down time user spends on browsing and checking the price of a product online daily. Our website will do this for the user and will notify the user on price drop of the products

**Problem Definition**

Many of us has faced the problem where we bought a product online after some month we notice there is a price drop on the product and we wish if there is a way to notify us on the price drop on the product before we had actually bought this product. Our website solves this problem and helps the user to keep track of the price of the product. And help to buy the product below the current price. This price drop in product price is generally during festive offers or big sales.

**Expected Outcome**

Expected outcome of our website is that it will be able to track the price of a product and notify the user on the price drop of the product. In our website the user will be able to add link to multiple product and track them simultaneously and will notify the use on the price drop of the product.

**Software Design & Requirement**

**Software Design:**

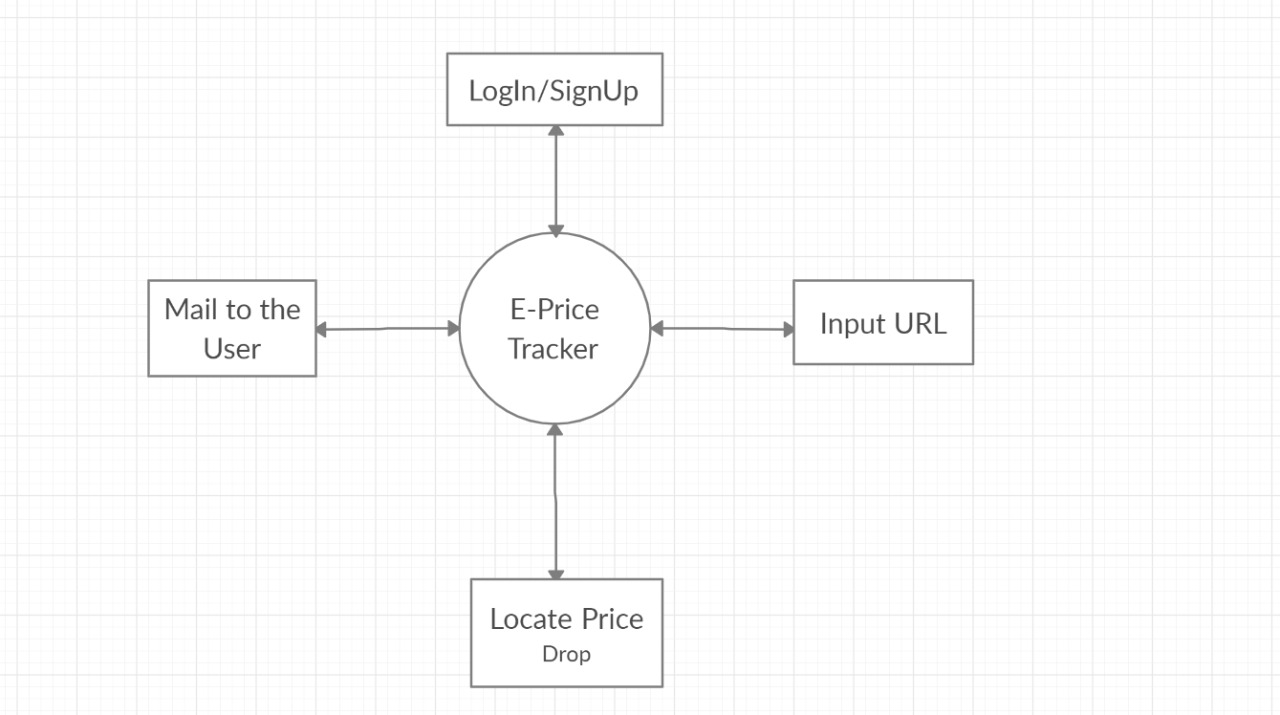
In this part, I have made use **Data Flow Diagrams, ER Diagram and Use Case Diagram** to create software and design.

A **data-flow diagram** is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

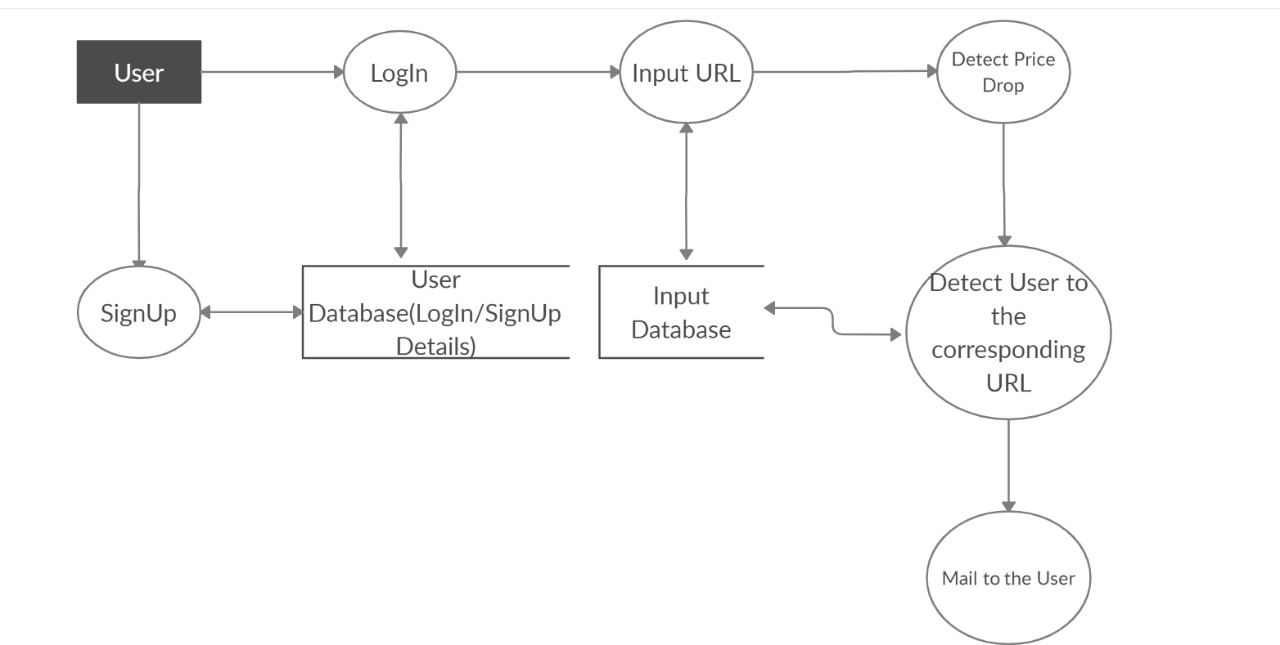
An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved in. A use case diagram can identify the different types of users of a system. The use cases are represented by either circles or ellipses.

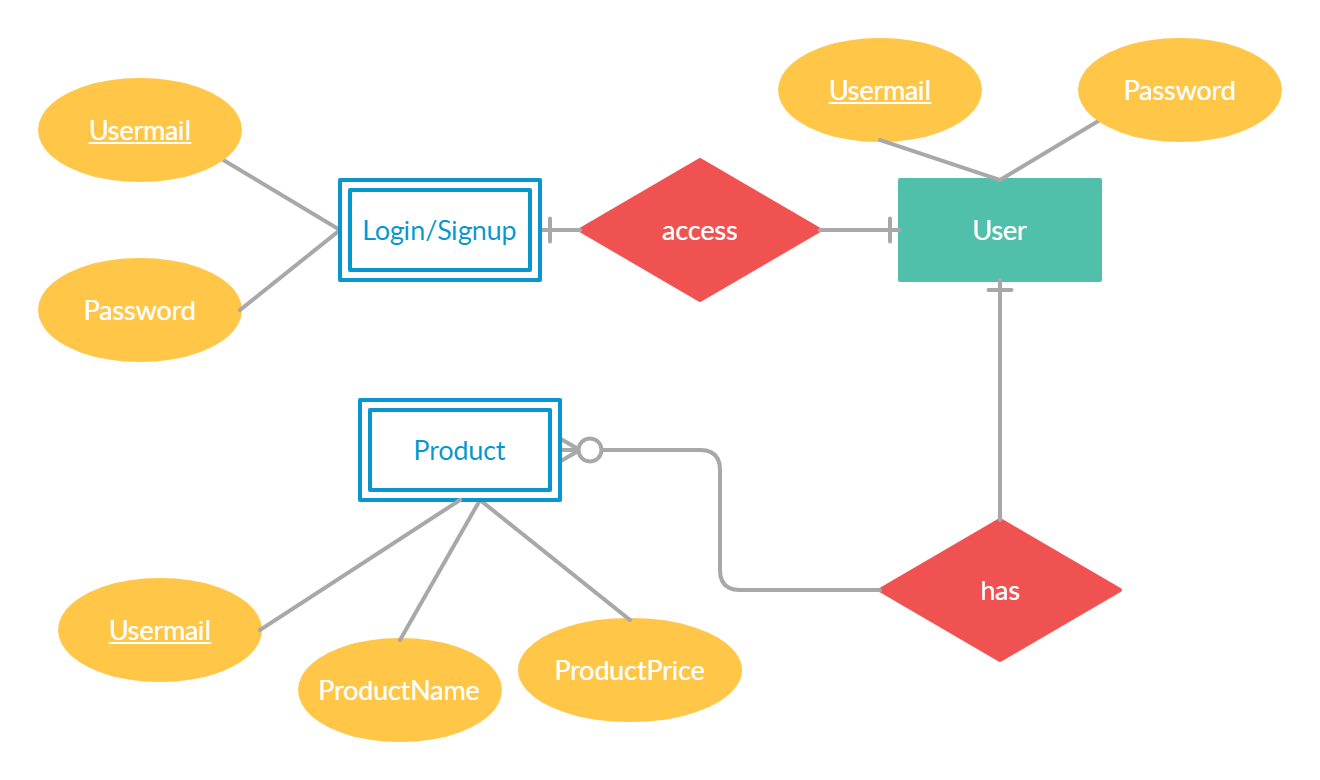
**DFD Level – 0**

****

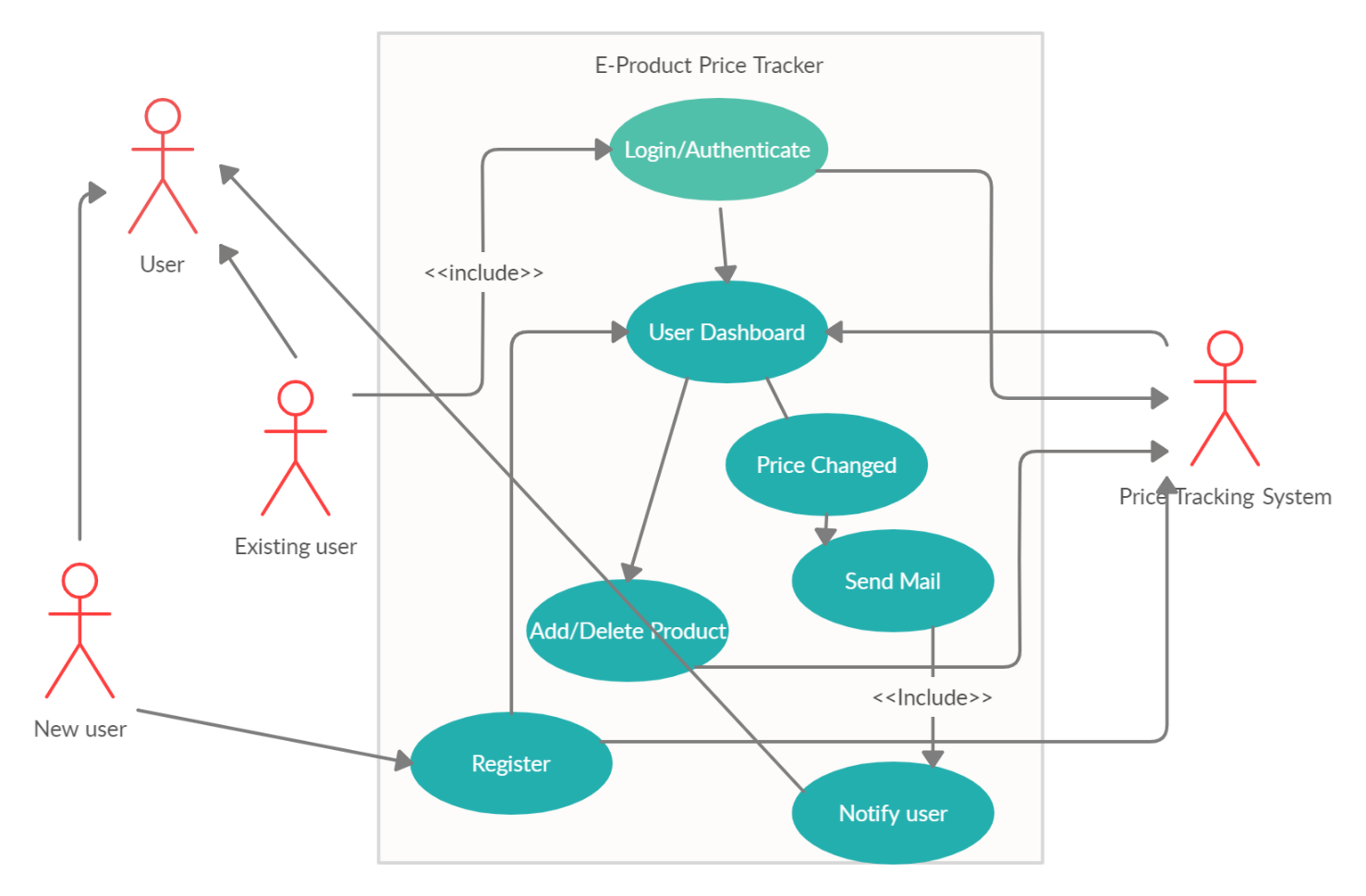
**DFD Level -1**

****

**ER DIAGRAM**

****

**USE CASE DIAGRAM**

****

**Requirement:**

Software:

* Visual Studio Code
* MongoDB Server

Hardware:

* Internet
* Computer with minimum i3 processor and 4GB RAM

**Technology Used**

❖ HTML

❖ CSS

❖ BOOTSTRAP

❖ JAVASCRIPT

❖ JQUERY

❖ NODE.JS

❖ EXPRESS.JS

❖ MONGODB

**Implementation Details**

****

**Front End:**

Front End is the area in the project that is visible to the user. This part is constructed using HTML, CSS and JavaScript. The structure of the website is developed using **HTML**, style is created using **CSS** and the behavior the website is developed using **JavaScript**.

Our **Front End** consists of a landing page which gives user a recommendation that if they are visiting our site for the first time then they could **SignUp** and if they have come to our website before then they could move to **LogIn** to our website. There are two forms in our project and users can fill in those forms with proper credentials, like their name, email and their password. These details will be saved in our database so that they could be used effectively on proper time.

**Back End:**

Back End is the area which is hidden from the user (Abstraction). We have used Node.js, Express, Embedded JavaScript (EJS) and MongoDB. **Node.js** we have used to execute our JavaScript code outside our web browser e.g. when we try to access a port on the server. **Express** is a minimal and flexible Node.js web application framework. It helps in fast tracking development of server- based applications. **EJS** is a simple templating language that lets us generate HTML markup with plain JavaScript. **MongoDB** is used to store the data in the database so that it could be used effectively later on.

We have Node.js to make an interaction between server and webpage using **Hyper** terminal command line, it makes the work easier as we have installed **NPM** packages and used **Nodemon** from it only**. Nodemon** gives the flexibility to our server to update automatically.

**Express** gave us the flexibility to use **EJS** which further gives us the flexibility to use it’s **‘view engine’**. Furthermore, it gave us the flexibility to make request and response using **callbacks**. The request and respond were made using **app.get()** method, where app contains express module. We store the express module into a variable so that it may have the module’s reference because express contains more than one module and by storing it can be used to create application.

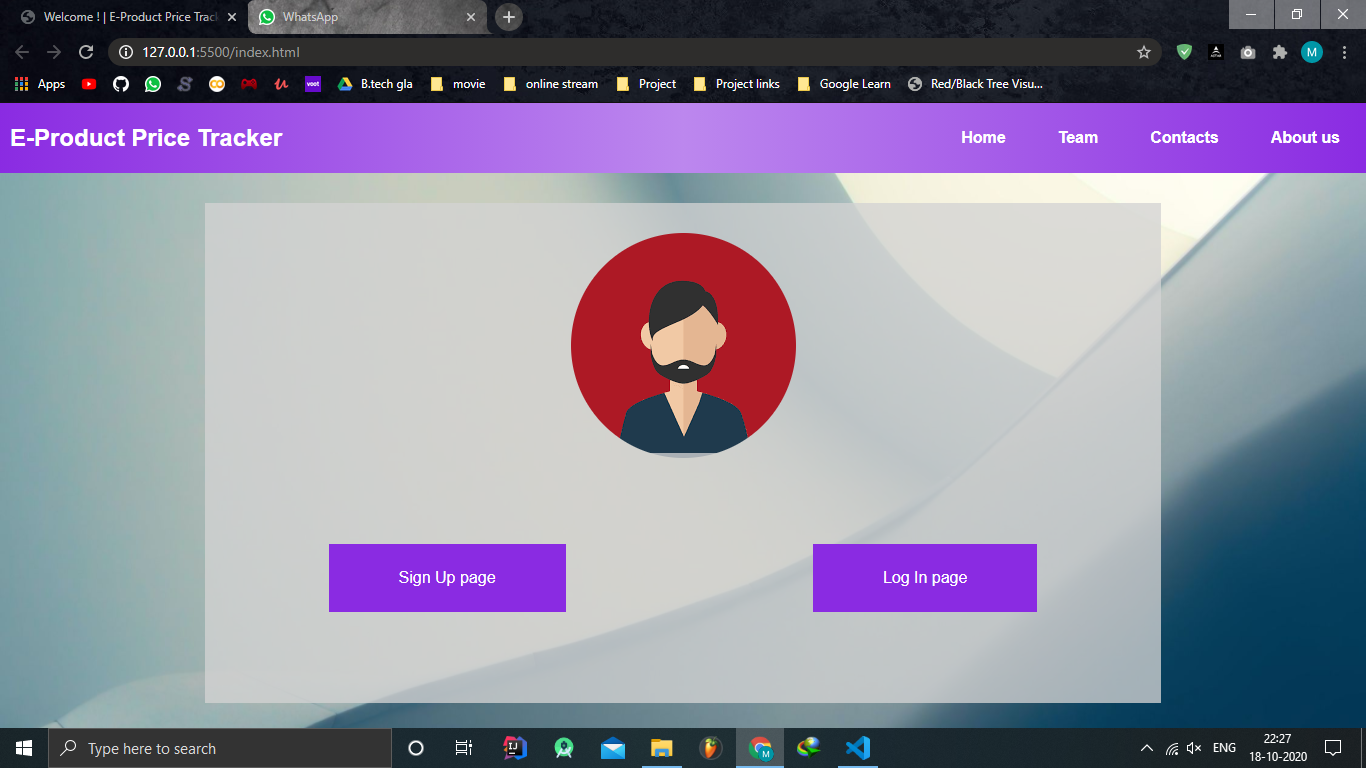
Then we used **EJS** for templating our HTML forms (signup and login) in which we have used the **post method** to send our data from our webpage to server. At the end we have used **MongoDB** so that our data may get stored at some place permanently and on further refreshing the page it may not get lost. The advantage of using mongo DB is that it supports a wide range of queries which can return user-defined JavaScript functions and regular expression searches.

**Progress till date & The Remaining Work**

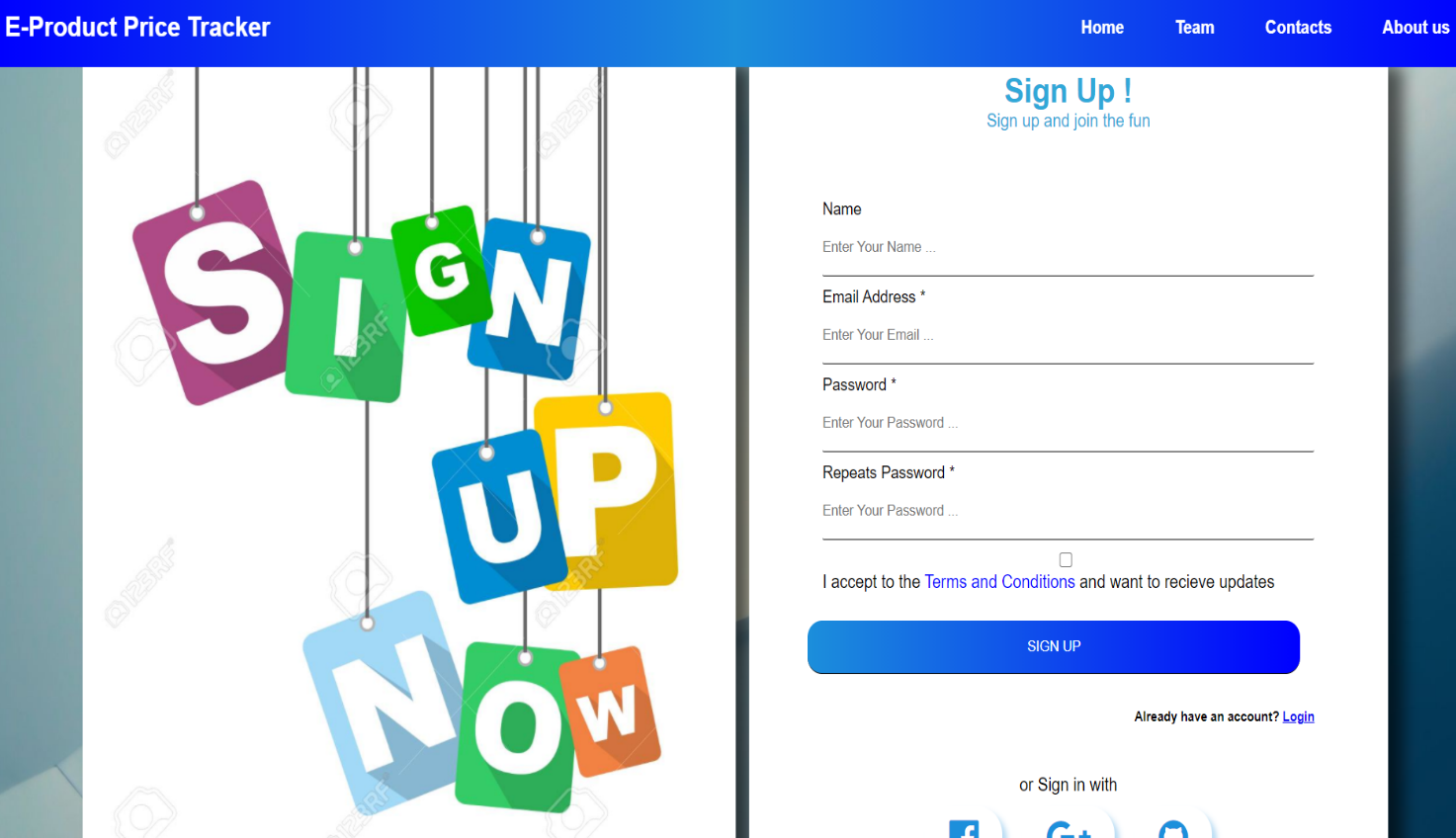
**Progress**

We have completed our Front End part with partial completion of out Back end part. Our Front End consists of our landing page which gives the recommendation to the user to signup or login to our website.

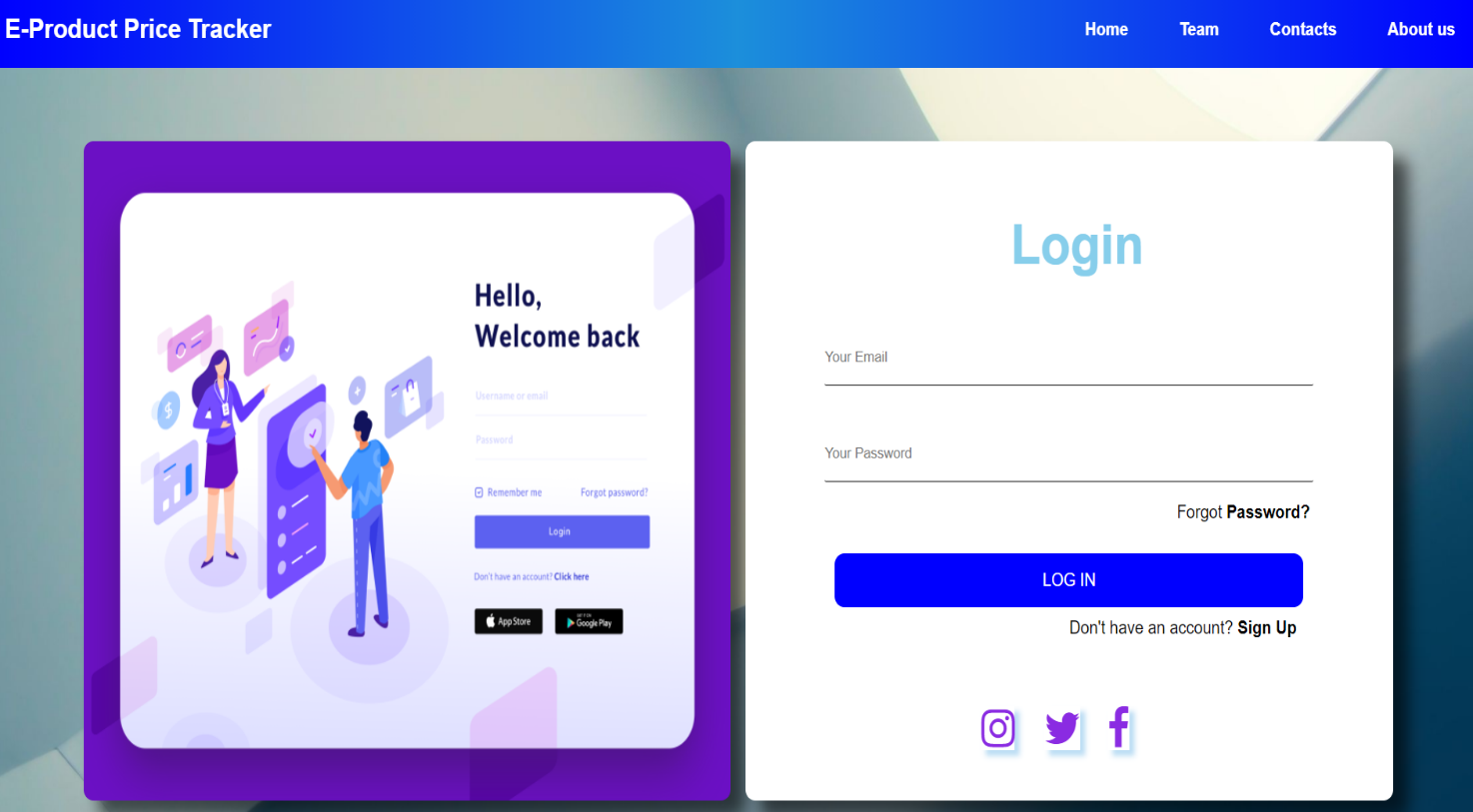
**Main/Landing Page:**



**Signup Form:**

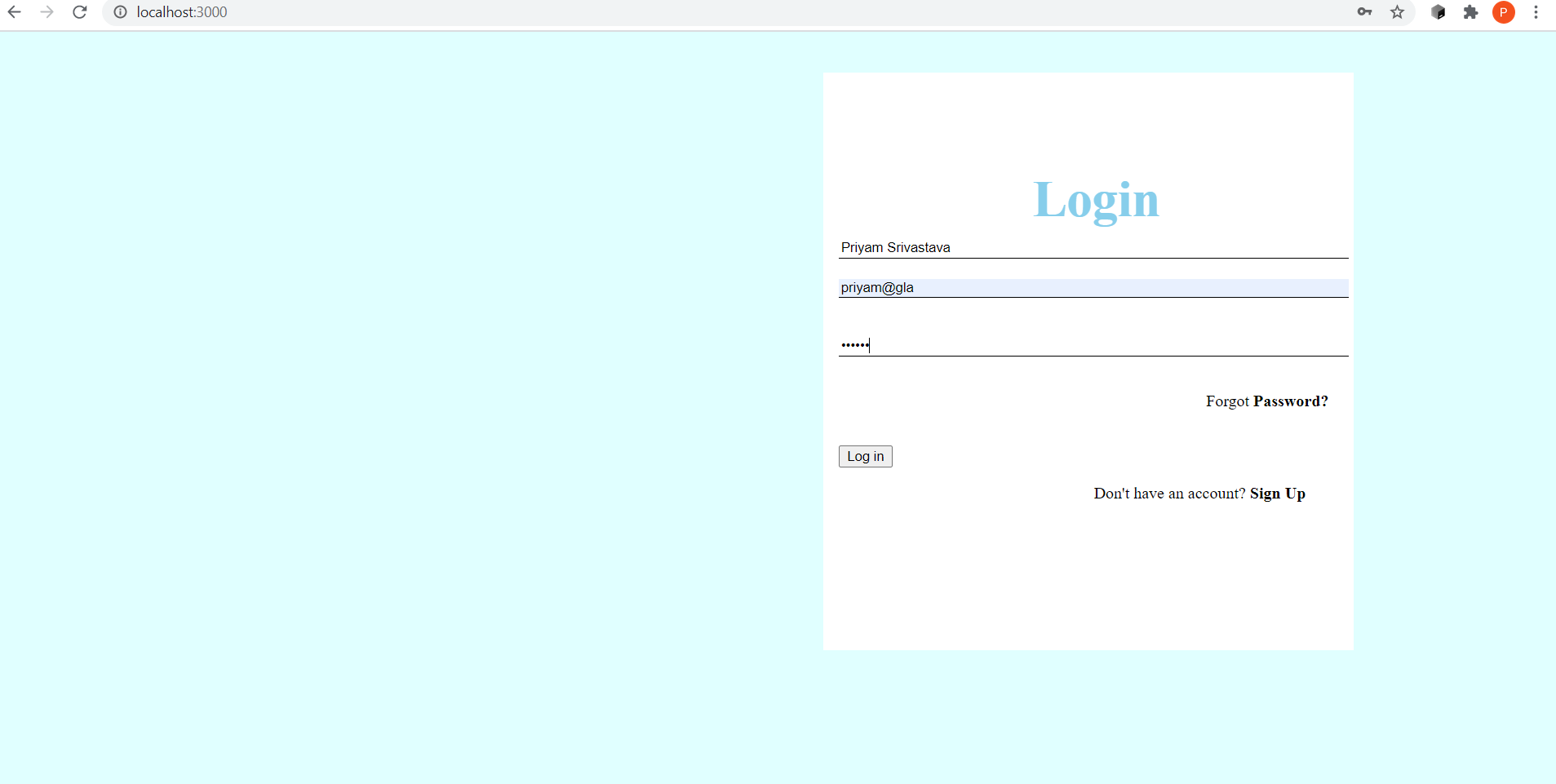
****

**Login Form:**

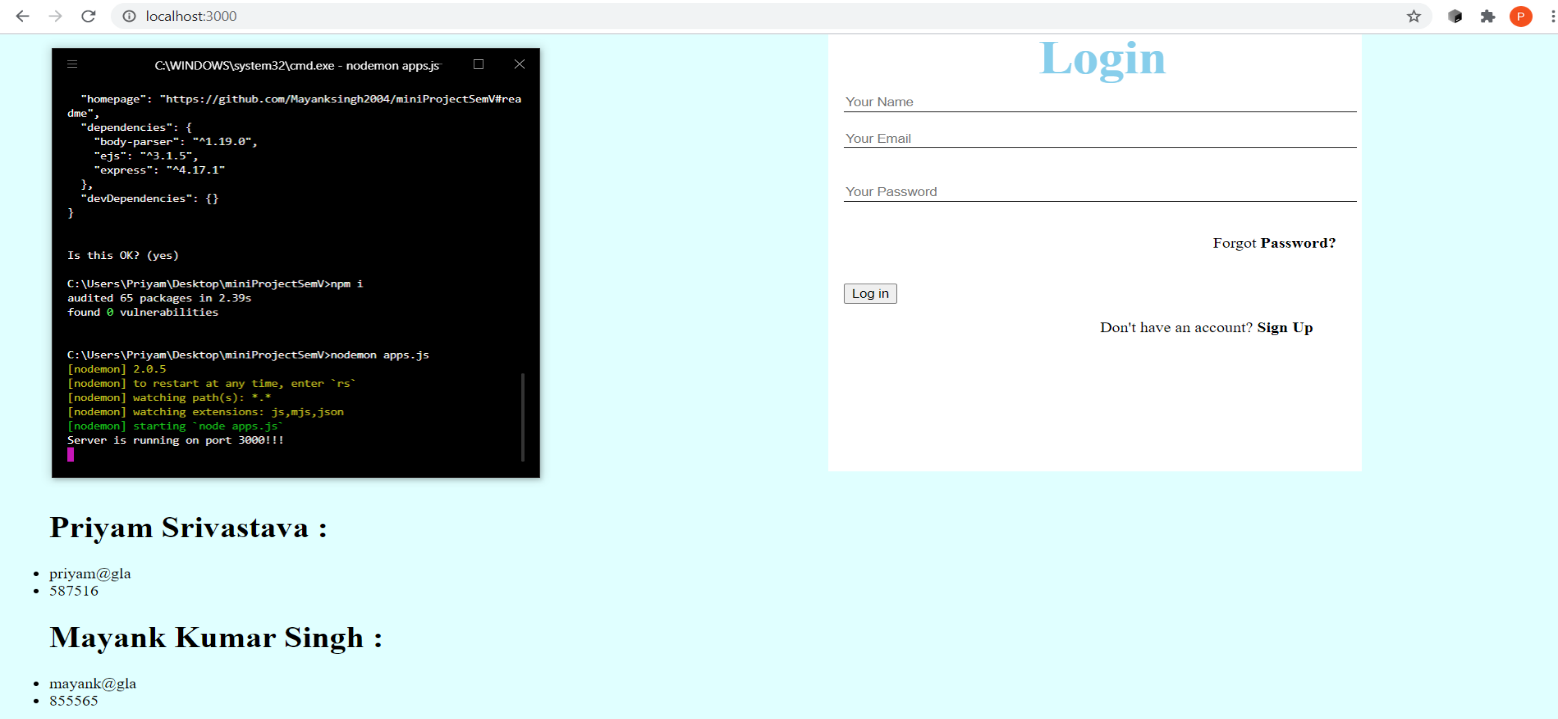
****

**Working on localhost: 3000**

This a way to input the data save on the server.

****

After the data is input then is gets submitted on the server side.

****

**Hyper Terminal:**

****

**Remaining Work:**

* **Connection with the Database:** Till now we have taken the input from the user and stored it on the server side but not connected it to the database.
* **Web Scrapping:** The scrapping of the data which user has filled in the form has to be scrapped and the relevant mail has to be send to the user on the base of the outcome of the scrapping.

**References**

* <https://medium.com/@martin.riedweg/amazon-price-tracker-with-zenaton-puppeteer-a6dbd3b9e174>
* <https://www.udemy.com/course/the-complete-web-development-bootcamp/>
* <https://w3school.org/>
* <https://stackoverflow.com/>