Auto Complete using Trie

Members:

Dhruva Khanwelkar (Roll no: 29, GR no: 11910603),

Harsh More (Roll no: 45, GR no: 11910933),

Nirvisha Soni (Roll no: 48, GR no: 11911362),

Juhi Rajani (Roll no: 55, GR no: 11910632),

Chirag Vaswani (Roll no: 74, GR no: 11911320)

- In this project, we aimed at making a web application using Express.js, JavaScript, HTML5, and CSS
- Express.js was used to write the backend of the website and the rest were used for front end as well as for implementing the Trie data structure.
- Our website had 2 fundamental functionalities.
    1. Registering a username
    2. Searching for that username
- The autocomplete feature was implemented on the front end of the website in JavaScript
- The username using which the user registered was added to a cookie using Express.js and then sent as a response along with an HTML page
- Then, the data from the cookie was retrieved and added to the Trie
- The Trie containing all the words was referenced and using a search function which we created
- We were able to make suggestions based on the data which the user has entered & populated the page with an unordered list containing all the suggestions
- The suggestions were just usernames as we stored just the usernames in the Trie

Here is the code from the file trie.js which contains the functions necessary for implementing the Trie data structure:

```
function makeNode(ch) {
  this.ch = ch;
  this.isTerminal = false;
  this.map = {}; // store a mapping from character
  this.words = []; // list of words
}
function add(str, i, root) {
  if (i === str.length) {
    root.isTerminal = true;
    return;
  }
  if (!root.map[str[i]])
    // check if the character we want to add already exists
    root.map[str[i]] = new makeNode(str[i]);
  root.words.push(str); // add the word to the list
  // console.log(root.words); -> Debugging
  add(str, i + 1, root.map[str[i]]);
}
function search(str, i, root) {
  if (i === str.length) return root.words;
  if (!root.map[str[i]]) return [];
  return search(str, i + 1, root.map[str[i]]);
}
```

- makeNode function is used to make nodes which are stored in the Trie
- add function is used to add nodes to the Trie
- search function is used to search for words from the Trie

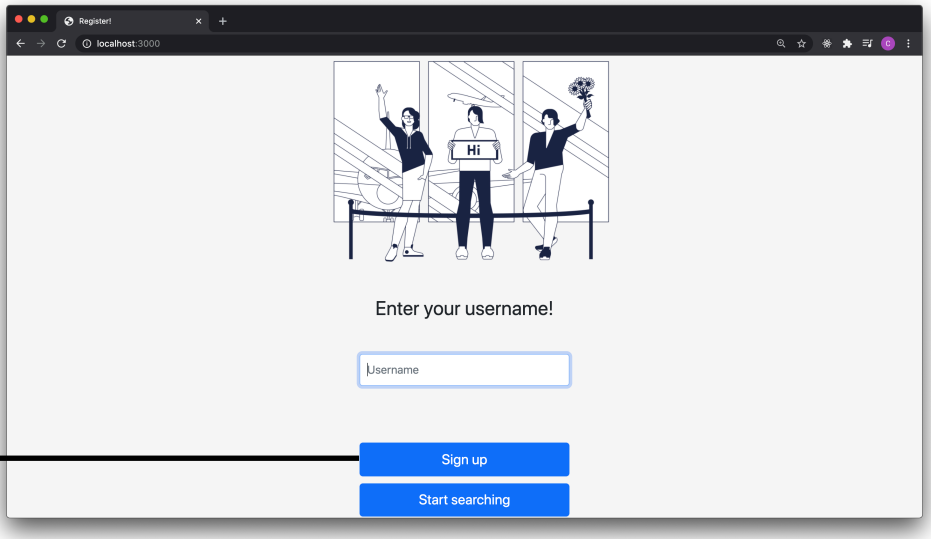
File containing the Express.js application looks like this:

```
const express = require("express");
const bodyParser = require("body-parser");
const app = express();
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static("static"));

app.get("/", (req, res) => {
  res.sendFile(__dirname + "/form.html");
});

app.post("/", (req, res) => {
  res.cookie(req.body.username); // Setting the cookie
  res.redirect("/search");
});

app.get("/search", (req, res) => {
  console.log(res.cookie);
  res.sendFile(__dirname + "/index.html");
});

app.listen(3000, () => console.log("Server running on
port 3000"));
```

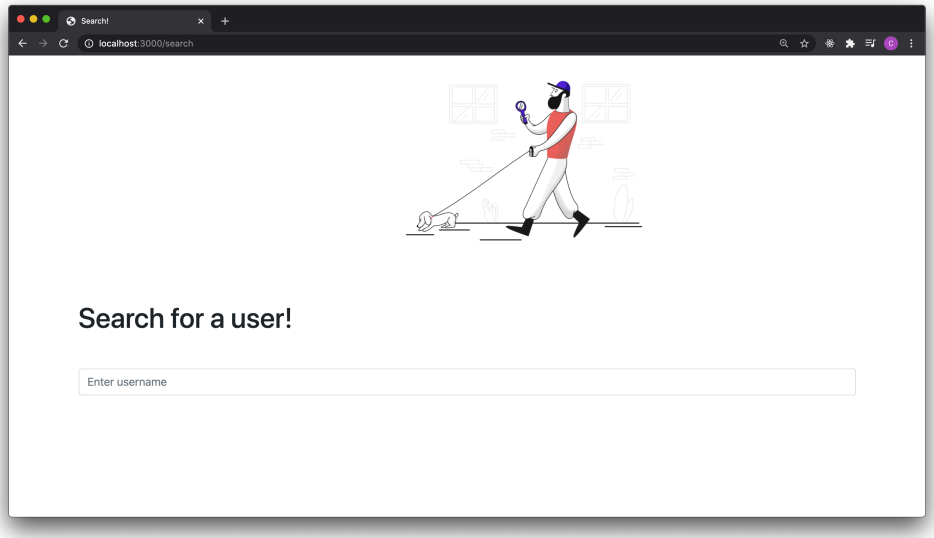
The code for the entire web application can be found at:

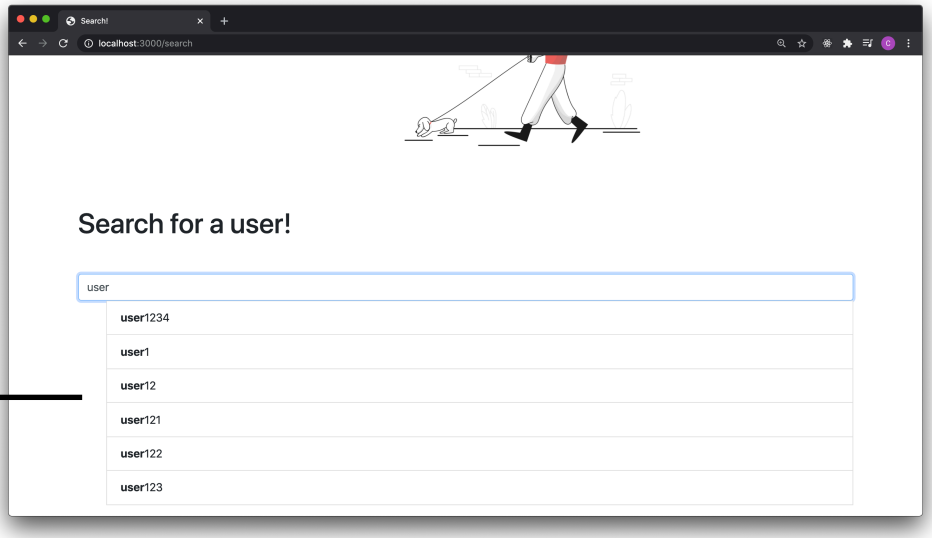
https://github.com/chiragvaswani/AutoComplete.git

Snapshots -



Registers users.
Redirects to /search





Search suggestions