Job Board Platform

# Dinesh Perni

[pernidinesh@cityuniversity.edu](mailto:pernidinesh@cityuniversity.edu)

Osama Khan

[khanosama@cityuniversity.edu](mailto:khanosama@cityuniversity.edu)

Umutbek Abdimanan Uulu

[abdimananuuluumutbe@cityuniversity.edu](mailto:abdimananuuluumutbe@cityuniversity.edu)

CS628 Full stack development, Master of Computer Science (MSCS), City University of Seattle

**Abstract**

The job market is highly competitive, especially for freshly graduated individuals who lack real-time experience but possess enthusiasm and a strong network of connections. To address this challenge and create a platform exclusively tailored to the needs of these young professionals, we propose a Job Portal project. The Job boarding project aims to empower human resources workers to post job opportunities specifically targeted at individuals who have graduated within the past two years. By focusing on this niche, the platform seeks to bridge the gap between employers seeking fresh talent and young graduates eager to kickstart their careers. Key features of the Job boarding include an intuitive and user-friendly interface that simplifies the job posting process for recruiters. Employers can highlight their preference for fresh graduates, ensuring that only relevant applications reach them, reducing the time and effort spent on sorting through a plethora of unrelated resumes. For fresh graduates, the platform will offer a personalized experience, tailoring job recommendations based on their educational background, skills, and connections. In addition, users can access a dynamic dashboard with key data, view a list of people in the system, and establish their own unique job categories. With category and location-based filtering options and a pagination function for simple navigation, the frontend provides a seamless job search experience. In this paper, the Job Portal's architecture, implementation, and functionality will be described in detail, with an emphasis on how quickly posting jobs and conducting job searches might be made more efficient. This project intends to describe the creation of a Full stack Job Portal using the MERN Stack, which will consist of React, MongoDB, Node.js, and Express.js. Through these methods, we hope to deepen our programming knowledge in MERN stack.

**Keywords:** Programming, Full Stack development, MERN Stack, React, Node.js, MongoDB, Job portal, Fresh graduates

**Introduction**

This project addresses an issue faced by millions of students around the globe by providing a simple solution that reduces the number of variables that are taken into consideration when designing and building a career portal. The project targets the unnecessary hurdles faced by fresh graduates where a fresh graduate gets mingled up in the vast pool applicants in a typical career portal website like Linkdin and Indeed and other alike our intention is to provide a platform where fresh graduates get equal opportunity with themselves and not with people who have experience and or are overqualified for that position that is set for a fresh graduate. Throughout the building phase of this project we hope to practice what we are taught in class and put in use the our learning and possibly create a practical solution to address the problem that we are aiming to solve.

**Background**

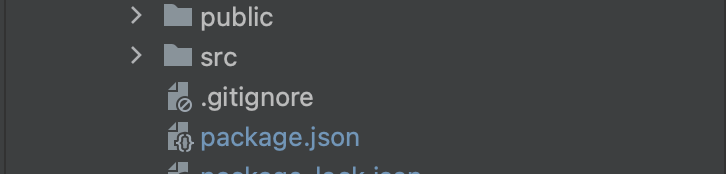
The group is at a point in their studies where they must start applying for jobs and are looking everywhere to get their hands on an opportunity that could land them a suitable job. Not to go very far in the past but the idea for this project aroused taking most recent circumstances in consideration that a typical fresh graduate would face. All of us at one point in our lives face issue like this and in today’s competitive market standing out from people with years of experience is not possible unless you compete on your own level. There are people switching careers and in doing so they have the managerial experiences of decades, but they present themselves as fresh graduates when they change their field of study which should not be the case, having experienced situations like these the group thought putting the idea into action.

**Methodology**

**Frontend**

Frontend refers to the part of a software application or website that interacts directly with the user. It is the user interface (UI) and the experience that comes with it (User Experience, UX). In web development, the frontend usually consists of everything the user experiences directly: text, images, buttons, layouts, animations, and any interactions. It contrasts with the backend, which refers to the server-side components of an application or website.

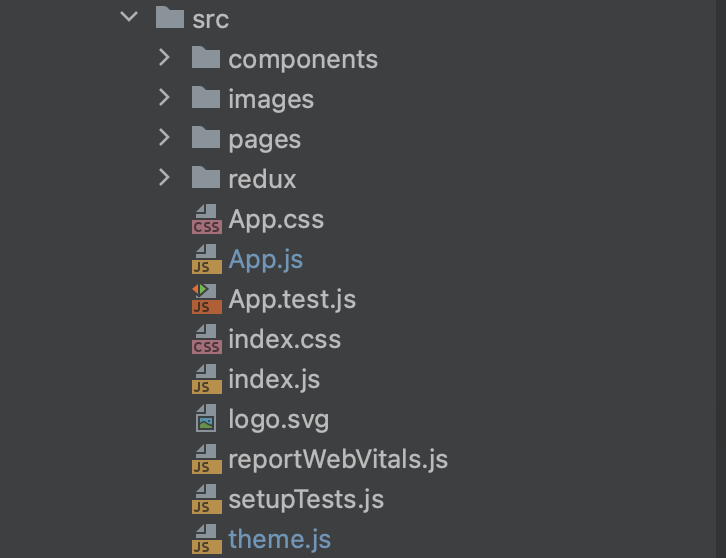
When creating a web application, it is recommended to logically separate the codes into directories. Firstly, let’s talk about Structure of Frontend



**public** - the index.html file and general information about the project.

**package.json** file contains information about the external libraries included in the project.

**src** - writes executable code and saves media files (shown below)



**components** - records components (folders and keys) that have been used more than once.

**Images** – used images

**pages** - contains pages. Uses components and containers inside.

**redux** - Writes the logic for saving, modifying, disabling and accessing a web application.

**index.js** file describes the integration of the responsex and the react library.

**Used software and tools**

**React.js**

React is an open-source JavaScript library developed by Facebook, primarily used for building user interfaces in single-page applications. It allows developers to create reusable UI components and manage the state of an application efficiently. React operates on a virtual DOM, enabling fast updates and rendering by minimizing direct manipulation of the actual DOM. Its component-based architecture and robust ecosystem have made it one of the most popular frontend frameworks in the web development community.

**Npm package manager**

NPM is a package manager for the JavaScript language. This will help you download the required packages easily and conveniently. NPM also creates its own package.json file. This file contains the names and versions of the required packages. With this file, other users can install all packages using the same npm install command.

**Redux framework**

Redux is a status management library for JavaScript applications. It helps to write stable applications that run in different environments (client, server) and are easy to test. Redux is ideal for medium to large applications. It should only be used when it is not possible to manage a standard situation in React or another library with a manager.

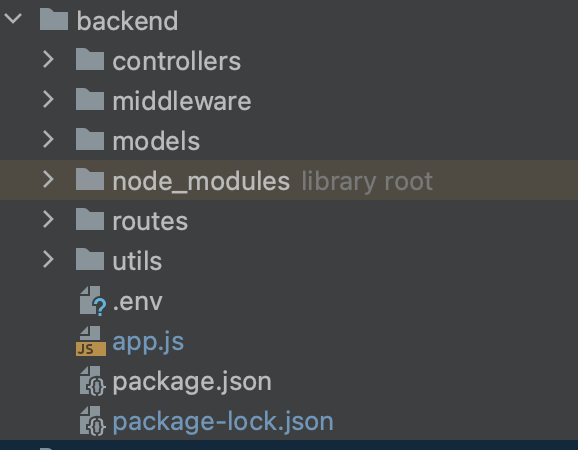
**Material-UI (MUI): A Comprehensive React Component Library**

Material-UI (MUI) is a popular open-source React component library that provides a set of pre-built components following the Material Design guidelines developed by Google. Material Design is a design language that emphasizes simplicity, clean lines, and a modern, visually appealing aesthetic. MUI enables developers to create responsive, interactive, and visually consistent user interfaces by offering a wide range of components and utilities.

**Backend**

In a software design project backend work is to manage and create server-side components, create process to efficiently run them through handling the logic, manage data storage and the interaction with databases. The backend takes in requests from frontend and based on the logics and database connection provides appropriate solution.

The most important thing for Beckend is the creation of the endpoint. Let’s talk about structure of NodeJS backend



**controllers**: This directory contains the controller files. Controllers handle the application's logic and interactions between the models and views. They receive requests from the routes, process data from models, and send responses back to the client.

**middleware**: Middleware functions are functions that can be executed before or after the request reaches the route handler. They can perform tasks such as authentication, logging, input validation, etc.

**models**: This directory contains the model files. Models are used to define the structure and behavior of the data. They usually interact with the database or data storage, and they represent entities in your application.

**routes**: In this directory, you'll find the route files. Routes define the endpoints that your application exposes to handle HTTP requests. They map incoming requests to specific controller functions.

**utils**: The "utils" directory often contains utility files that contain reusable functions or modules that can be used across different parts of your application.

**.env:** This is a configuration file where you can store environment-specific variables, such as database connection strings, API keys, and other sensitive information.

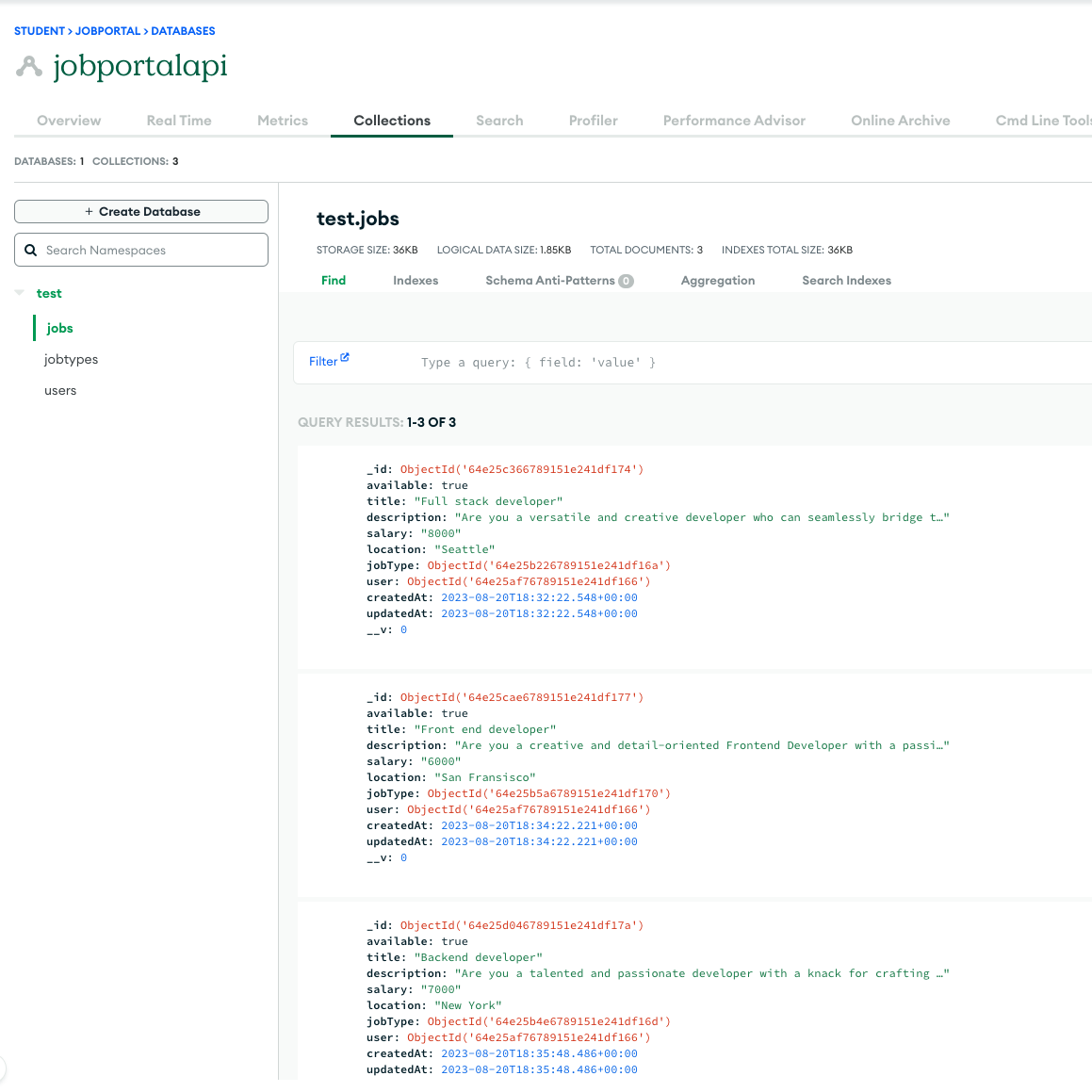
**app.js:** This is the entry point of your application. It's where you set up your server, configure middleware, define routes, and start listening for incoming requests. It's where the various components of your application come together.

**package.json:** This file contains metadata about your application, including its name, version, dependencies, and scripts. It's used by npm (Node Package Manager) to manage your project's packages and scripts.

**Database – MongoDB**

MongoDB is a popular open-source NoSQL database management system that is designed to handle large volumes of unstructured or semi-structured data. It is particularly well-suited for modern application development where flexibility, scalability, and quick development cycles are essential. When used in conjunction with Node.js, MongoDB offers several benefits that align well with the characteristics of Node.js applications:

MongoDB's flexibility, scalability, and compatibility with JavaScript align well with the asynchronous, event-driven nature of Node.js. The combination of these technologies can lead to more efficient development, streamlined data management, and the ability to build high-performance real-time applications.



**Other used software**

**- Postman**

In this project we used postman for API testing. Postman is an application used for API testing. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated.

Postman offers many endpoint interaction methods. The following are some of the most used, including their functions:

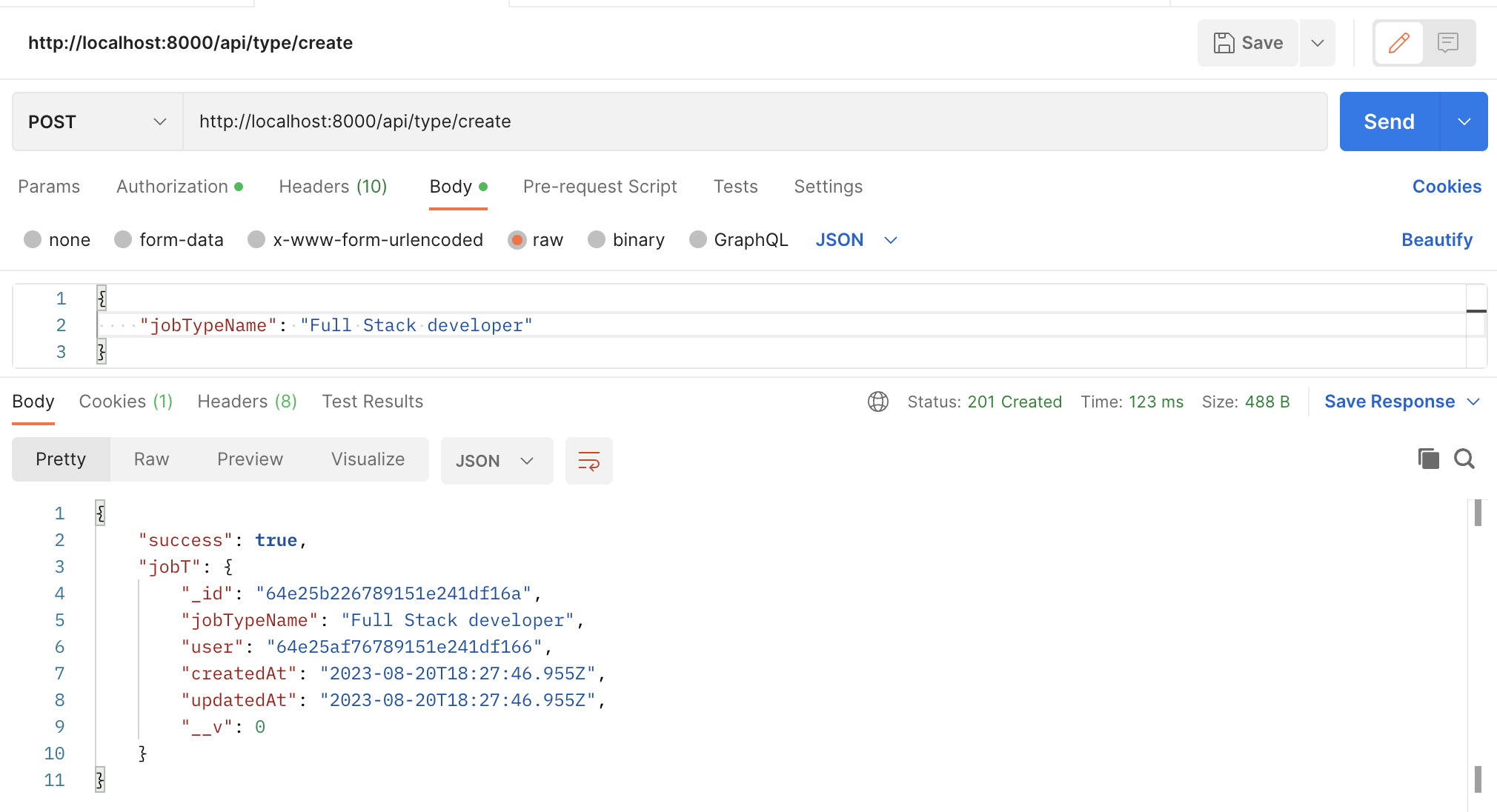
- GET: Obtain information

- POST: Add information

- PUT: Replace information

- PATCH: Update certain information

- DELETE: Delete information



**- Github (** [**https://github.com/Umutbek/TP\_CS628**](https://github.com/Umutbek/TP_CS628)**)**

GitHub has played a pivotal role in driving the growth of open-source software development. It provides a platform for developers to freely share their work, collaborate with others, and contribute to a wide range of projects, ranging from small utilities to large, complex software applications. It has helped democratize software development by making it more accessible and encouraging collaboration across borders and backgrounds.

**Application idea**

The app has two parts:

* User part (freshers)
* Administration part

**1. User Part:**

**Sign Up & Profile Creation:** Fresh graduates can sign up using their email or through our website. Users fill in details such as educational background, skills, achievements, preferred job categories, location preference, and a brief bio.

**Job Search & Filtering:** Users can search for jobs based on keywords, job category, company name, and location.

**Job List:** Can see list of Jobs

**Application Process:** Directly apply for jobs with their profile or upload a specific resume.

Users can track the status of their applications (e.g., applied, shortlisted, interviewed, rejected).

They can see list of applied jobs

**2. Administration Part:**

**Employer Sign Up & Profile Creation:** Companies can sign up and create their organizational profile, highlighting their culture, values, and the benefits of joining them.

**Job Posting:** A straightforward interface for recruiters to post job openings, specifying details like job description, qualifications required, salary range, etc.

Specify that the position is for fresh graduates to get relevant applications.

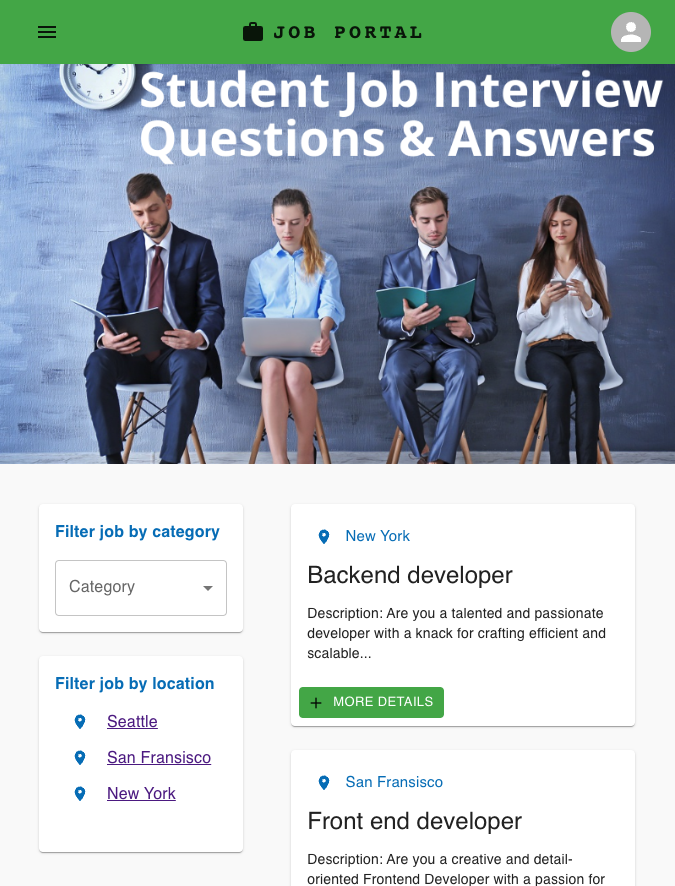
**Applicant Tracking System:**

View, sort, and manage the received applications. Shortlist, schedule interviews, or send rejection emails right from the platform.

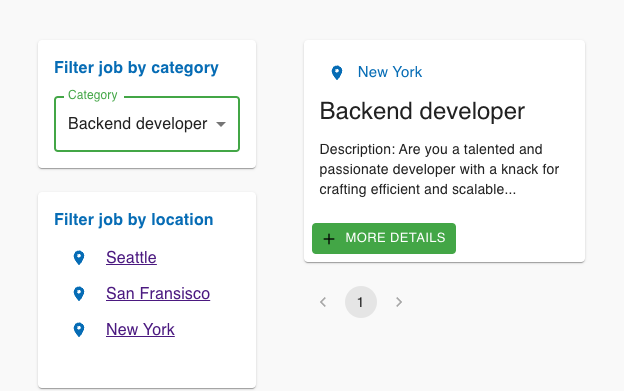
**Analytics Dashboard:** Track the number of views on a job post, the number of applications received, sources of applications, and more.

**Interface Description**

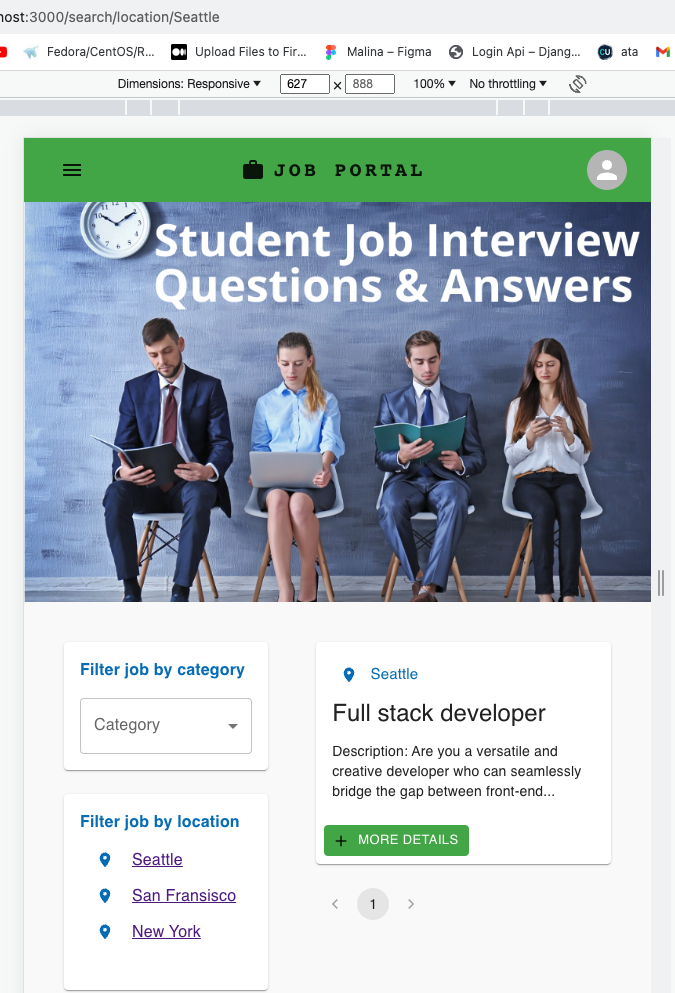
**Main page:** User will be redirected to this page, when he/she entered the website. You can see list of available jobs, filter it by categories and location. To view more about job he needs to click MORE DETAILS button.



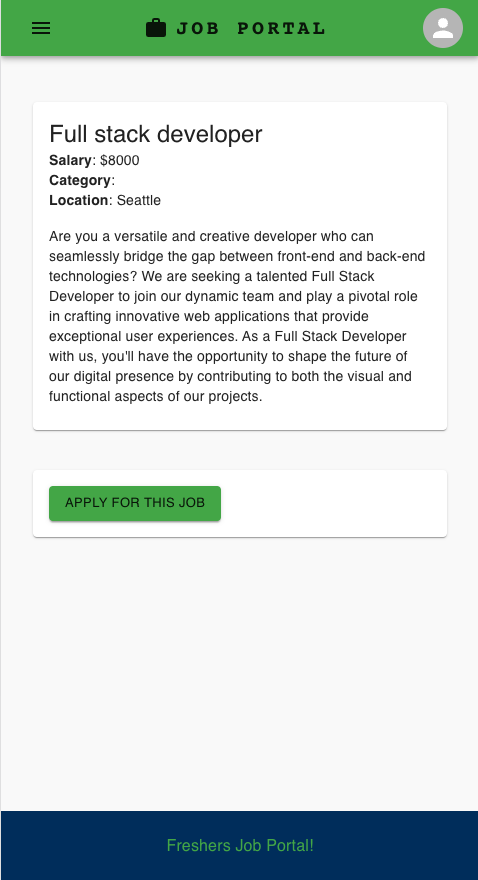
User can filter jobs by category and locations



Filter by location ( Seattle selected )



If user clicks to More Details, he will be redirected to this page.



To apply for a job and get access to the Dashboard user have to login system first. Currently this parts are in progress. It’ll be available for TP03

**9.** **REFERENCES**

Srivastava, S., & Srivastava, S. (2023). How to Create A ReactJS Developer Job Description: Hiring Guide For Global Recruiters. *Uplers*. <https://www.uplers.com/blog/reactjs-developer-job-description/>

Svitlana Varaksina (2023). How to Create a Job Board Website Like Glassdoor

<https://themindstudios.com/blog/create-a-job-board-website/>

CHATGPT