A red letter on a black background

Description automatically generated

DSCI-D 532 Applied Database Technologies

Job Tracker

Shalini Kothuru (skothuru@iu.edu)

April 28, 2024

1. **Application Link:**

We hosted out project on the React platform, using its free tier services. Since it is a free tier, the website is slow to load as it becomes inactive if not used continuously, and has the potential to crash with heavy traffic,

Website\_URL: <https://adt-final-project.onrender.com>

1. **GitHub Link:**

We have uploaded all the required code files, template files, and all project related documents on GitHub in the following repository,

GitHub\_Repo\_URL: <https://github.iu.edu/skothuru/JobTracker> (Initial)

<https://github.com/shalinikothuru/JobTracker(updated> (updated to personal account)

1. **Project Summary:**

Our project is all about making job hunting easier. We are building an application that keeps track of job applications, so you don't have to. It helps you stay organized, follow up on job applications, and see what's happening in your job search.

Youtube Link: <https://www.youtube.com/watch?v=x3e0aee5ex0>

A yellow sign with black text

Description automatically generated

1. **Objectives:**

The primary objective of the Job Tracker website is to offer users a comprehensive and user-friendly platform to efficiently manage and track their job applications. This includes the job application submission dates, the status of each application, and follow-up actions. The website aims to help job seekers by providing them with valuable insights through visualizations in the interface and metrics related to their job search process, thereby enabling them to make informed decisions and optimize their job search strategy. Additionally, the website seeks to facilitate better organization and time management by allowing users to schedule and manage follow-up reminders for communication with companies, recruiters, or interviewers, ensuring that users can maintain proactive engagement with their job applications.

1. **Usefulness:**

The usefulness of the Job Tracker Log application lies in its ability to provide a centralized platform for job seekers to organize and manage their job application process effectively. We have listed some of key aspects of its usefulness below:

**1. Enhanced Organization:** Users can maintain a detailed record of all job applications, including dates of application and the status of each application, in one place. This eliminates the need for manual tracking methods, such as spreadsheets or physical notebooks, making the job search process more streamlined and less overwhelming.

**2. Insightful Analytics:** By offering interactive user interface representations and metrics of the user's job application efforts, the website enables users to identify trends, and areas for improvement in their job search strategy. This can include insights into which industries or job roles they are having more success in, and how their application volume correlates with interview requests.

**3. Time Management:** By organizing and tracking all job applications in one place, users can save time and reduce the stress associated with keeping track of multiple application timelines and deadlines manually.

**4. Accessibility and Convenience:** Being an online platform, Job Tracker allows users to access their job application records from anywhere at any time, providing flexibility and convenience for job seekers who are often on the move or managing their search alongside other commitments.

Overall, the Job Tracker website is designed to make the job application process more efficient, less stressful, and more successful for job seekers by providing the tools and insights needed to manage and optimize their job search efforts.

1. **Dataset**

**6.1 Tables**

The database for this project is going to be user generated, and consists of the of two (2) tables –

1. **‘Users’** Table – to keep track of the user demographic data. This table will have the following columns –
   1. **‘User\_id’ –** This column will keep track of the unique userid created by the user at the time of account creation.
   2. **‘Email’ –** This column will keep track of the unique emails used by the users at the time of account creation.
   3. **‘First\_name’ –** This Column will store the user’s First Name associated with the account.
   4. **‘Last\_name’ –** This Column will store the user’s Last Name associated with the account.
   5. **‘Password’ –** This Column will store the user password associated with the account.
   6. **‘Gender’ -** It indicates the gender of the column.
   7. **‘Account\_created’ –** This Column will store the Date & Time of the account creation for every user.
   8. **‘Contact\_No’ -** The mobile number of the user given by user at time of signup.
2. **‘Jobs’** Table – to keep track of the variety of job positions applied to by the user.
   1. **‘Id’ –** It is the primary key of this table.
   2. **‘Job\_id’** - Stores a unique or non-unique identifier assigned to each job posting, allowing for the aggregation of multiple records associated with the same job. This facilitates tracking different stages or applications of the same job posting over time.
   3. **‘User\_id’** - the id of the user. It will be automatically retrieved by the user session.
   4. **‘Company\_name’ –** The name of the company or organization name to which user has applied.
   5. **‘Role’ –** The position in the company that user has applied. Ex- Data Scientist, Data Analyst etc.
   6. **‘Location’ –** This column will store the location of the position, eg – Miami, Remote, etc.
   7. **‘URL’ –** This column will store the web URL for the job position, providing easy access and retrieval for the job post for a user.
   8. **‘Job\_source’ –** A column to keep track of the portals used for applications (eg – LinkedIn, Indeed, etc.). This column is for the users analytical purposes to determine which portal has the highest count of applications.
   9. **‘Referral’ -** It is used to indicate if the person has applied using referral.
   10. **‘Status’ –** This column will store the status of the job application, which can be updated as and when a user receives any updates regarding the application. It contains integer value to indicate the phase like submitted, rejected, interview, offer.
   11. **‘Application\_date’ –** This is the date on which the user has applied. User mentions this date.

**6.2. Data types, keys and Constraints**

The following constraints have been added to the selected columns for the above tables. The constraints might be updated later if required –

**Users Table:**

|  |  |
| --- | --- |
| **Column name** | **Data Type/key/Constraints** |
| User\_id | INT – Autoincremented value,  Primary key – It serves as a primary key of this table |
| Email | VARCHAR (50), UNIQUE, NOT NULL Constraints to ensure there are no multiple accounts for the same user |
| First\_name | VARCHAR (100), NOTNULL |
| Last\_name | VARCHAR (100), NOTNULL |
| Password | VARCHAR (25), NOTNULL,  Front End constraint - to verify length is more than 8 characters. |
| Gender | CHAR (1), NOTNULL |
| Account\_created | DATE, It stores current\_date at time of the account creation. |
| Contact\_NO | VARCHAR (10) NOTNULL, CHECK length of contact number is 10. |

**Jobs Table:**

|  |  |
| --- | --- |
| **Column name** | **Data Type** |
| Id | INT – Primary Key, Autoincremented value |
| User\_id | INT – FOREIGN KEY – Users (User\_id) |
| Job\_id | VARCHAR (50), NOT NULL |
| Company\_name | VARCHAR (50), NOT NULL |
| Role | VARCHAR (50) NOT NULL |
| Location | VARCHAR (25), NOT NULL |
| URL | VARCHAR (255) |
| Job\_source | VARCHAR (20), NOT NULL |
| Referral | VARCHAR (3), Default value is NO, i.e not applied by referral |
| Status | VARCHAR (20), NOT NULL |
| Application\_date | DATE, NOT NULL |

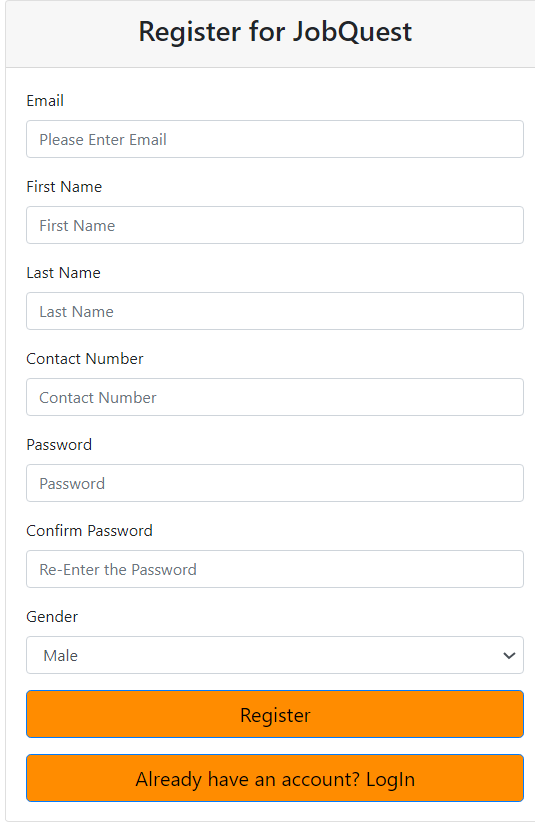
1. **Tools and Technologies used:**

We used Flask for the front-end development of our project, leveraging its simplicity and flexibility. Flask facilitated URL routing, dynamic HTML generation through Jinja2 templates, serving static files, and form handling via Flask-Form.

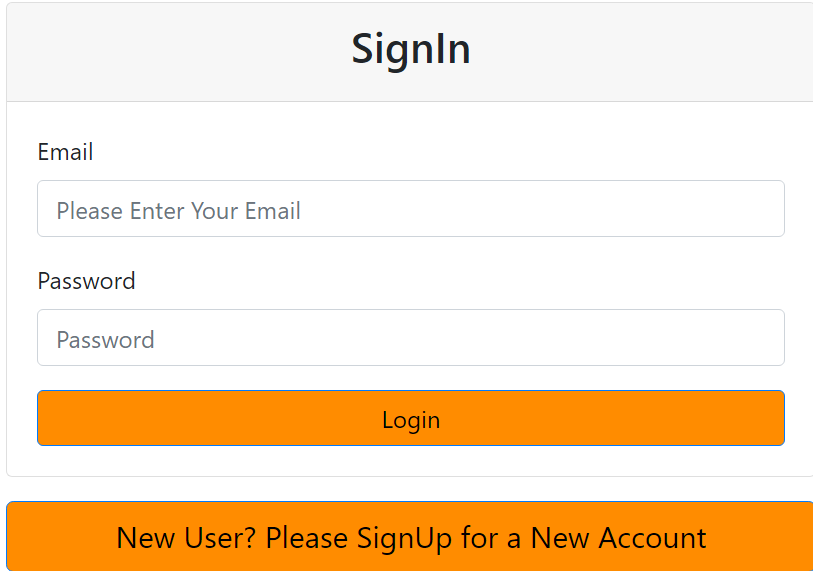
For the database, we employed SQLite due to its lightweight nature and efficiency. SQLite allowed us to define database schemas, perform CRUD operations, ensure data integrity through transactions, and seamlessly integrate with Flask using extensions like Flask-SQLAlchemy.

1. **Functionalities:**
2. **User Authentication:**

* **Sign up:** Users can create a new account by providing details like name, email, contact number, password, and gender. The password needs to be confirmed for verification.

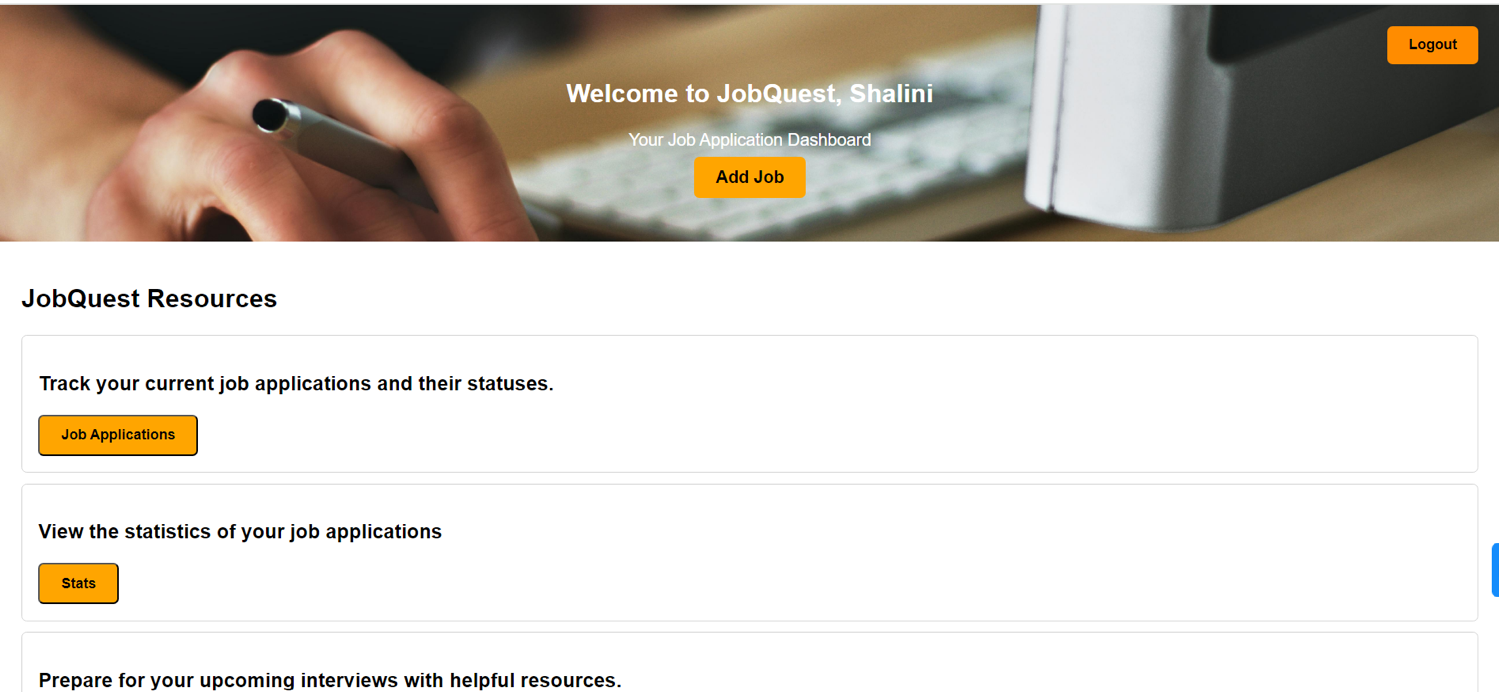


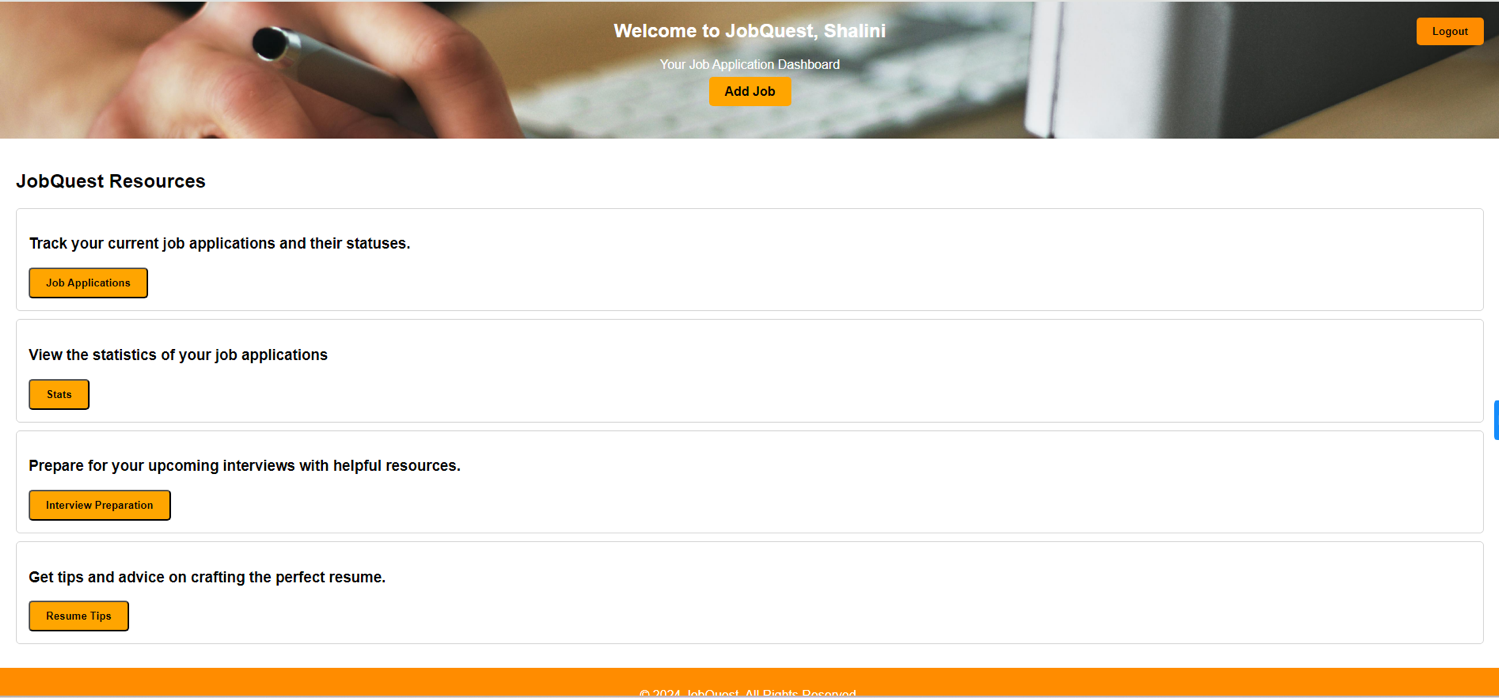
* **Login:** Existing users can log in using their email and password.



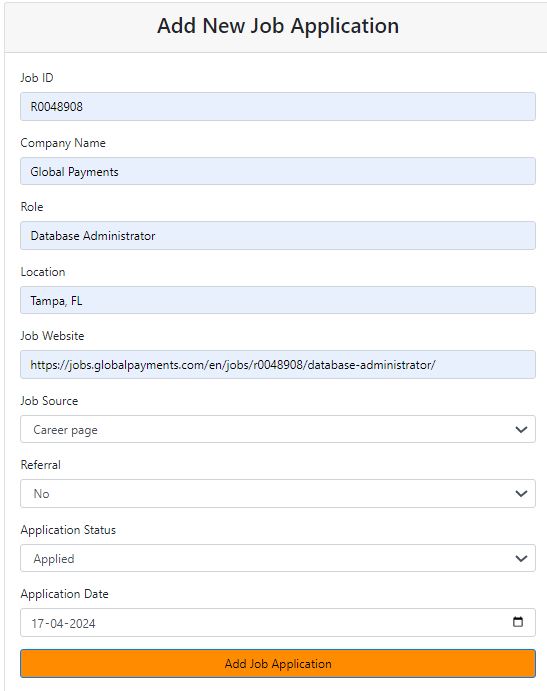
1. **Dashboard:**

The landing page of dashboard welcomes user and contains different buttons to add job application, view job applications, stats, resume tips, interview preparation, logout functions.

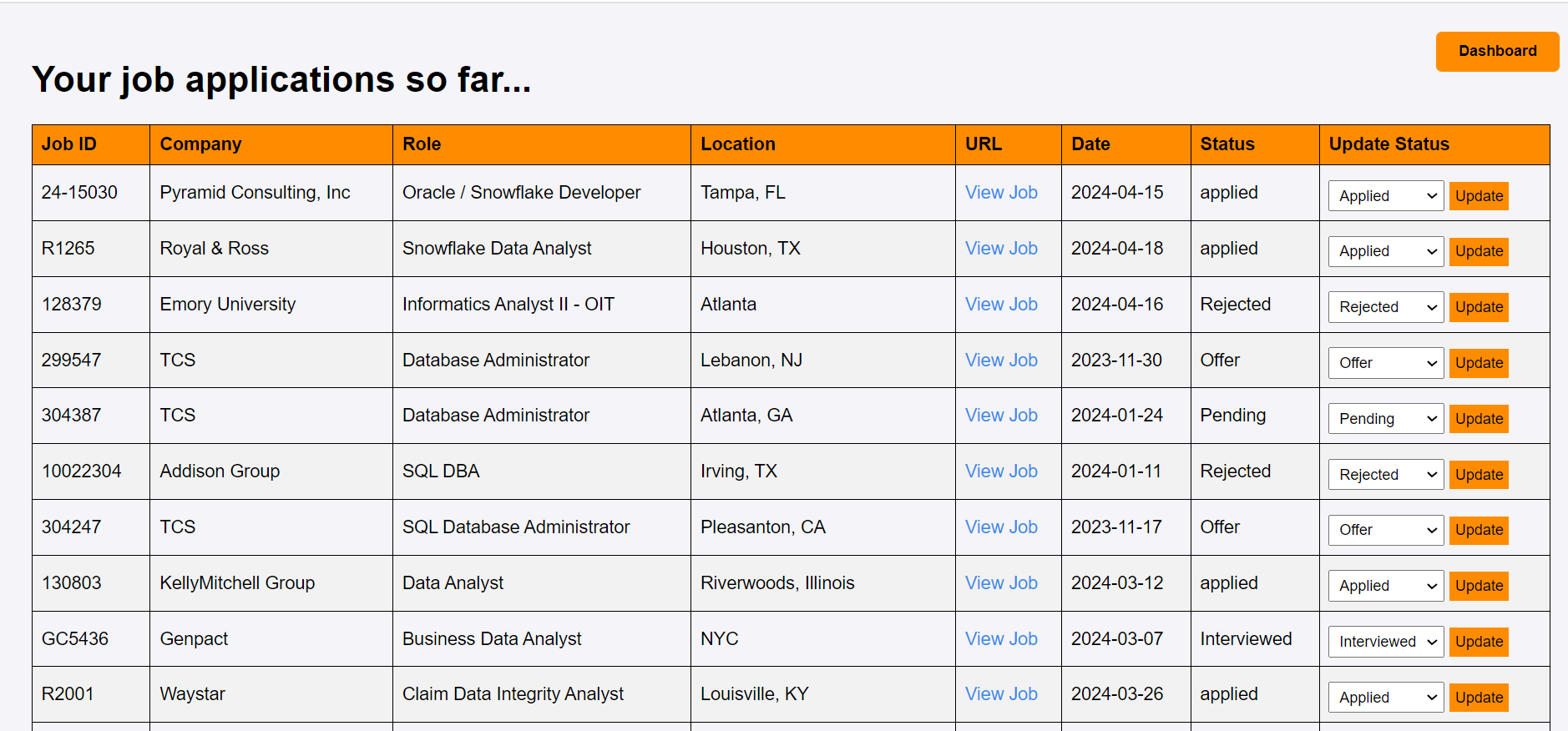




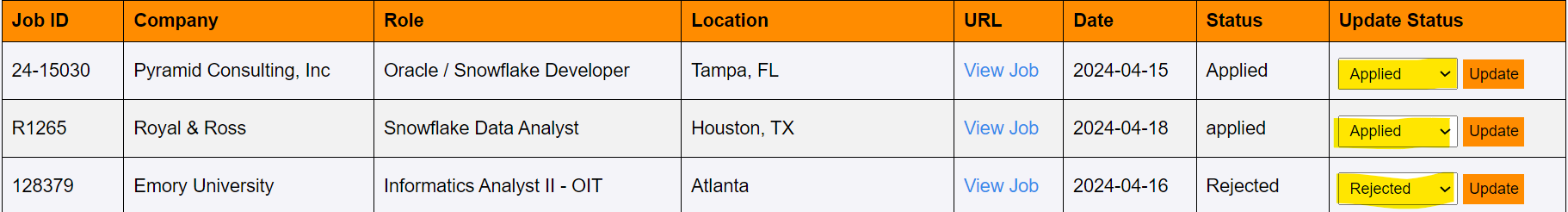
* **Add New Job Details:** Users can add details of a new job application including company name, role, location, job website, job source, whether it's referred or not, application status, and application date. These fields are useful for discovering insights like applications by different job board etc



* **View Current Job Applications:** Users can see a list of their current job applications along with their statuses.

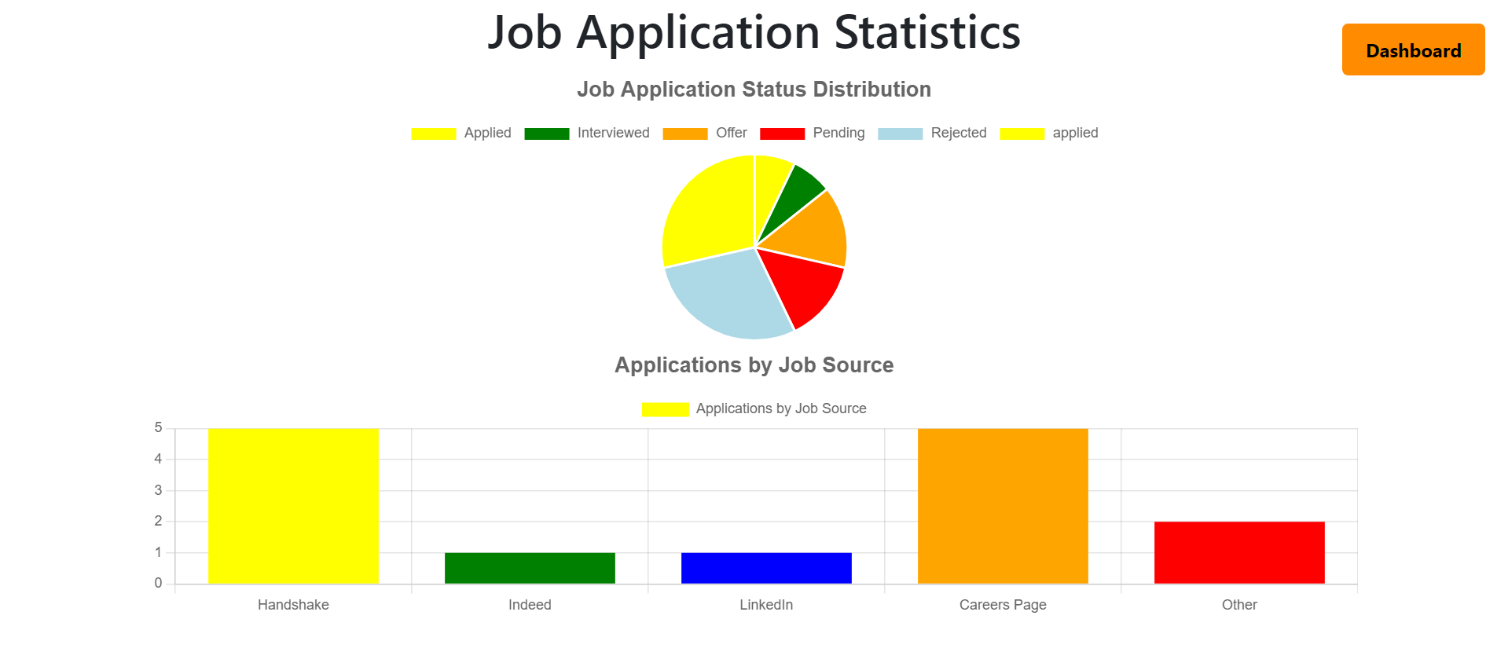


* **Modify Job Status:** Users can update the status of their job applications such as applied, interviewed, offer, pending, or rejected. It will be a dropdown. Dropdown is not visible while taking the screenshot of the page.



1. **Stats Page:**

* **Visualization of Job Information:** Users can view visualizations such as graphs or charts representing their job applications' data, like application status distribution, application date trends, etc.

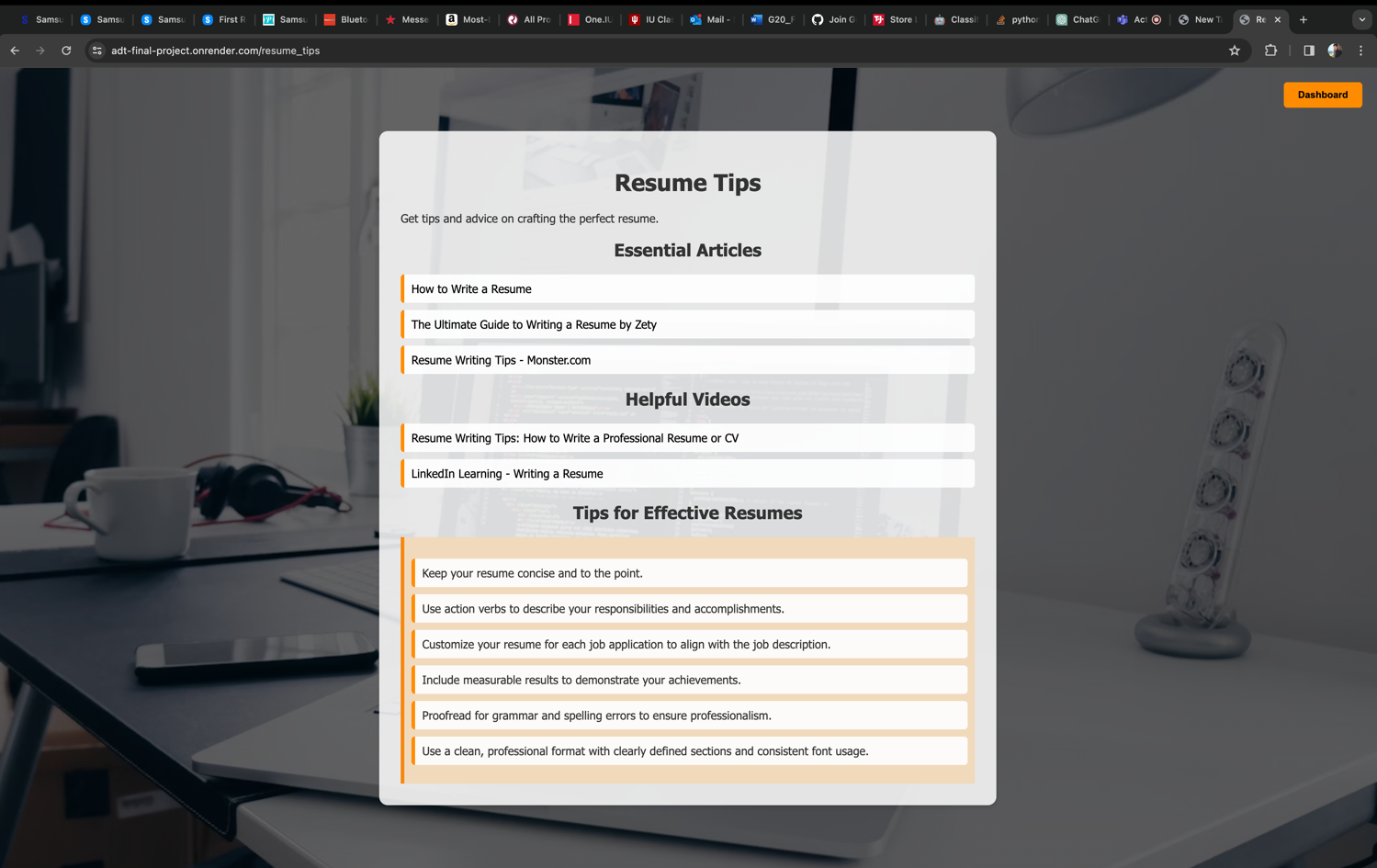


1. **Interview Tips and Resume Preparation:**

* **Access Useful Links:** Users can access links to resources for interview tips and resume preparation. These resources could include articles, videos, or templates to help users improve their resume and prepare for interviews.

A screenshot of a computer

Description automatically generated



1. **Navigation:**

* **Home Page:** Users are initially directed to a home page where they can choose to sign up or log in.
* **Dashboard:** After logging in, users are directed to the dashboard where they can manage their job applications and access other features. Also, users can redirect from any page to dashboard by clicking on dashboard from the respective page.
* **Add Job:** From Dashboard, user can click on add job to add their new job application.
* **Job applications:** Users can view their applications by clicking Job applications from dashboard.
* **Stats**: Users can view the insights from their job applications by clicking on the stats button on dashboard.
* **Interview Tips and Resume Preparation Pages:** Users can navigate to pages containing resources for interview tips and resume preparation from the dashboard.

1. **Security:**

* User passwords are securely stored using encryption techniques.
* Authentication mechanisms are in place to ensure that only authorized users can access their account and data.

1. **Challenges Faced:**

The primary challenges we encountered were in integrating the SQLite database with our Flask application and hosting it on Render. To solve the database connection issue, we did some online research and found a helpful YouTube video that guided us through the process.

During the hosting phase, our application, hosted via a GitHub repository, encountered deployment problems because it couldn't access the correct template files. We resolved this by updating the code to ensure smoother deployment and addressing the errors. A YouTube video served as a valuable resource for debugging and offered a general guide on the correct deployment procedure.