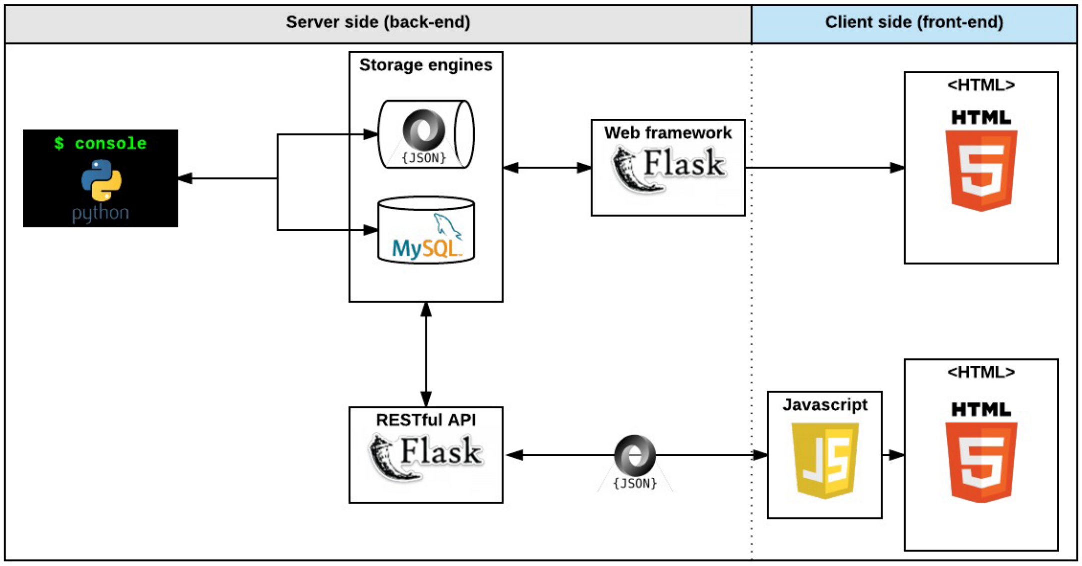
CareerLink

Web Architecture :



We will employ a client-server architecture.

The diagram illustrates the end-to-end flow of data in the job board MVP. The web client communicates with the web server through a RESTful API. The web server interacts with a database to store and retrieve data. Here's an overview:

1. Front-End (Web Client):

   - User Interface (UI): Built using HTML, CSS, and JavaScript.
   - Front-end Framework, Flask and Java Script: Provides a responsive and interactive user interface.
   - User Registration and Login Components.
   - User Profile Management Components.
   - Job Listing Pages.
   - Job Search and Filtering Components.
   - Application Submission Forms.
   - User Dashboard Components.
   - Notifications System (may use WebSocket or other technologies for real-time updates).

2. Back-End (Web Server):

   - Web Application Framework, Flask:

    - Handles HTTP requests and routing.
    - Implements middleware for authentication and validation.
- User Authentication and Authorization:
    - Manages user accounts and sessions.
    - Utilizes tokens for user sessions.
- RESTful API Endpoints:
    - User Management: Registration, Login, Profile Editing.
    - Job Listings: Create, Read, Update, Delete (CRUD) operations.
    - Job Search and Filtering.
    - Application Submission.
    - User Dashboard Data Retrieval.
    - Notifications Handling.
- Data Validation and Sanitization: Protects against security vulnerabilities.
- Integration with Third-Party Services: Email notifications and push notifications
- Security Measures: Protection against common web application vulnerabilities.

3. Database:

    - Database System, MySQL:
        - Stores user profiles, job listings, applications, and other data.
    - Database Schema:
        - Tables for users, job listings, job history, profile info, notifications and applications, structured according to data model shown in the data model section.
        - Indexing for efficient data retrieval.

Features

1. SIgn up and Sign in: a section where existing users can log in and new users can create an account.
2. Landing Page(Dashboard): a section which will serve as a landing page. You'll get to see different job offers or better still, different work profile. ((I was thinking, what if there were 2 sets of dashboard, one for employers and one for employees. The one for employers will be showing different job seekers with their qualifications and that for the employees will be showing different job offers))
3. User profile: users can create detailed profiles with information like work experience, education, skills, and contact details.
4. Employee Job listing: Employers can post job listings with information about job titles, descriptions, requirements, and location. This section will show literally all about the job.The pay, the time and every other thing the user needs to know about the job.
5. Job Search and Filters - Implement a search bar to help job seekers find relevant job listings. Provide basic search filters, such as location, job type (full-time, part-time, contract), and industry.
6. Application Submission **-** Allow job seekers to apply for jobs directly through the platform. Include the option to upload and manage resumes and cover letters.

APIs and Methods

User Registration and Authentication:
Route: /api/user/register
Method: POST
Description: This route handles user registration, allowing new users to create accounts. The client will send user registration data to the server, such as username, email, and password.
Route: /api/user/login
Method: POST
Description: Used for user authentication, allowing existing users to log in. The client sends user credentials (email and password), and the server validates them.
POST /api/users/logout: Handles user logout.

Landing Page (Dashboard):
Route: /api/users/:id/dashboard/jobs
Method: GET
Description: Retrieves job offers and work profiles to display on the landing page. The client sends a request to get the list of job listings and work profiles.

User Profiles:
Route: /api/user/:id/profile
Method: GET
Description: Fetches the user's profile information. The client can use this route to retrieve the user's own profile data.
Route: /api/user/:id/profile
Method: PUT
Description: Allows users to update their profile information. The client sends updated user details to the server.

Employee Job Listing:
Route: /api/jobs
Method: POST
Description: Enables employers to post new job listings. The client sends job details, including job title, description, requirements, and location.
Route: /api/jobs
Method: GET
Description: Retrieves a list of job listings. The client uses this route to display available job opportunities.
PUT /api/jobs/:id: Allows employers to edit job listings.
DELETE /api/jobs/:id: Allows employers to delete job listings.

Job Search and Filters:

GET /api/jobs/search: Allows job seekers to search for job listings based on location and job type

GET /api/jobs/filters: Provides filter options (e.g., location, job type).

Application Submission
POST /api/applications: Allows job seekers to apply for jobs directly through the platform and upload resumes and cover letters.
GET /api/applications/:id: Retrieves application details.
GET /api/applications/job/:jobId: Retrieves applications for a specific job listing.

The following API endpoints or function/methods will be created to allow other clients to use:

- createUser(): Creates a new user account.
- loginUser(): Logs in a user.
- logoutUser(): Logs out a user.
- createProfile(): Creates a new user profile.
- updateProfile(): Updates an existing user profile.
- createJobListing(): Creates a new job listing.
- getJobListing(): Retrieves job details for a specific job listing.
- searchJobListings(): Searches for job listings based on location, job type, and industry.
- applyForJob(): Allows job seekers to apply for jobs directly through the platform and upload resumes and cover letters.
- getDashboard(): Retrieves user dashboard data.

Data Models:

The below diagram illustrates the data model for the job board MVP. The database consists of 5 tables: users, profiles, job_listings,job_history and applications.

- The users table stores user account information.
- The profiles table stores user profile information.
- The job_listings table stores job listing information.
- The applications table stores job application information.

- The job_history table stores job histories information for a user.

- **__users**

Column Name	Data Type	Description
user_id	integer	Unique identifier for each user
email	string	User’s email address
password	string	User’s password
account_type	string	User’s account type (job seeker or employer)

- **job_history**

Column Name	Data Type	Description
job_history_id	integer	Unique identifier for each job history entry
user_id	integer	Foreign key referencing the user who owns the job history
job_title	string	Job title
company_name	string	Company name

start_date	date	Start date of the job
end_date	date	End date of the job

- **job_listings**

Column Name	Data Type	Description
job_id	integer	Unique identifier for each job listing
employer_id	integer	Foreign key referencing the employer who posted the job
job_title	string	Job title
company_name	string	Company name
location	string	Job location
job_description	string	Job description
application_instructions	string	Application instructions

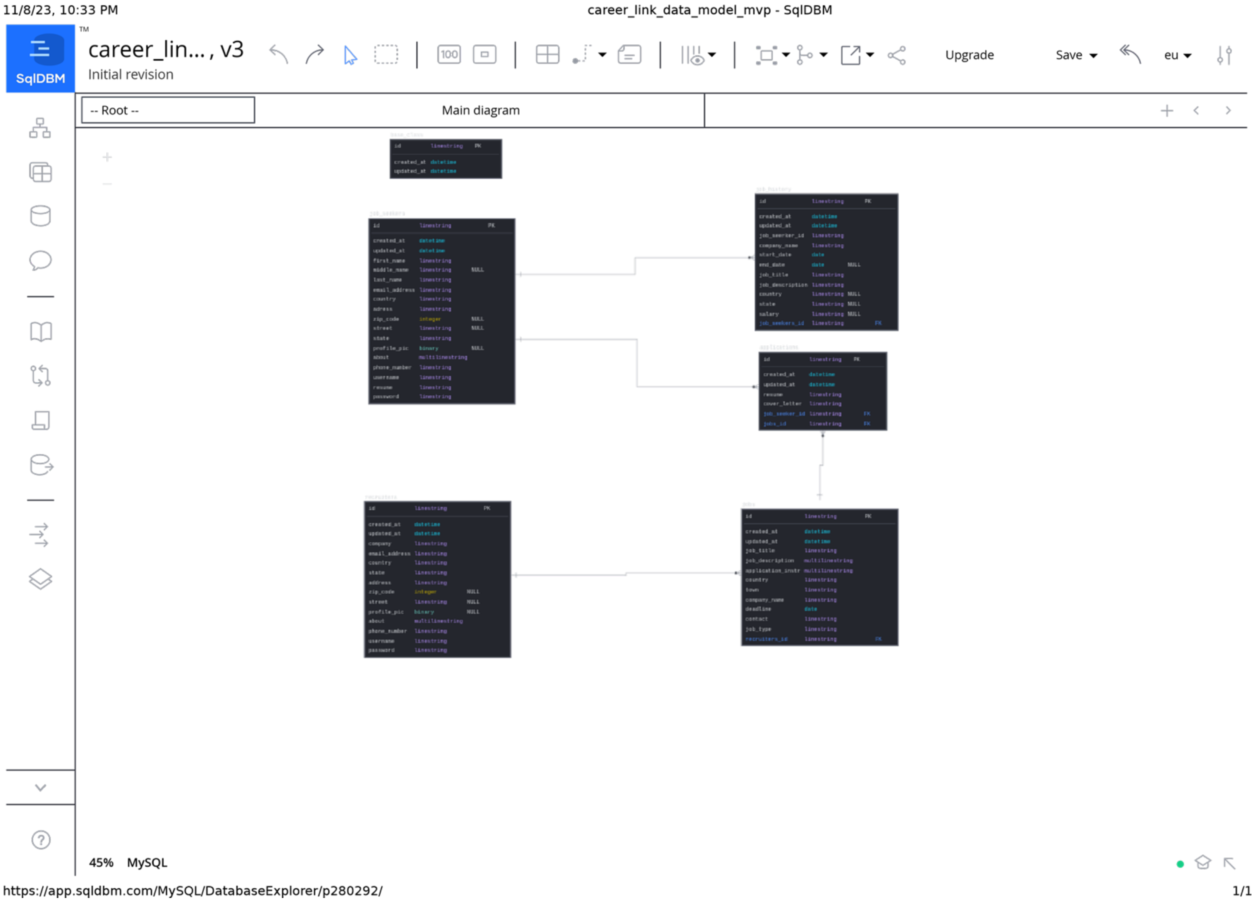
- **applications**

Column Name	Data Type	Description
application_id	integer	Unique identifier for each job application
job_id	integer	Foreign key referencing the job listing to which the application is submitted
user_id	integer	Foreign key referencing the job seeker who submitted the application
resume	string	File path to the job seeker’s resume
cover_letter	string	File path to the job seeker’s cover letter

- **profiles**

Column Name	Data Type	Description
profile_id	integer	Unique identifier for each profile
user_id	integer	Foreign key referencing the user who owns the profile
first_name	string	User’s first name
last_name	string	User’s last name

job_history	string	User’s job history



11/8/23, 10:33 PM career_link_data_model_mvp - SqlDBM

base_class

id	linestring	PK
created_at	datetime	
updated_at	datetime	

job_seekers

id	linestring		PK
created_at	datetime		
updated_at	datetime		
first_name	linestring		
middle_name	linestring	NULL	
last_name	linestring		
email_address	linestring		
country	linestring		
adress	linestring		
zip_code	integer	NULL	
street	linestring	NULL	
state	linestring		
profile_pic	binary	NULL	
about	multilinestring		
phone_number	linestring		
username	linestring		
resume	linestring		
password	linestring		

job_history

id	linestring		PK
created_at	datetime		
updated_at	datetime		
job_seerker_id	linestring		
company_name	linestring		
start_date	date		
end_date	date	NULL	
job_title	linestring		
job_description	linestring		
country	linestring	NULL	
state	linestring	NULL	
salary	linestring	NULL	
job_seekers_id	linestring		FK

applications

id	linestring	PK
created_at	datetime	
updated_at	datetime	
resume	linestring	
cover_letter	linestring	
job_seeker_id	linestring	FK
jobs_id	linestring	FK

recruiters

id	linestring		PK
created_at	datetime		
updated_at	datetime		
company	linestring		
email_address	linestring		
country	linestring		
state	linestring		
address	linestring		
zip_code	integer	NULL	
street	linestring	NULL	
profile_pic	binary	NULL	
about	multilinestring		
phone_number	linestring		
username	linestring		
password	linestring		

jobs

id	linestring	PK
created_at	datetime	
updated_at	datetime	
job_title	linestring	
job_description	multilinestring	
application_instr	multilinestring	
country	linestring	
town	linestring	
company_name	linestring	
deadline	date	
contact	linestring	
job_type	linestring	
recruiters_id	linestring	FK

https://app.sqldbm.com/MySQL/DatabaseExplorer/p280292/ 1/1

User stories:

Here are some detailed user stories that the job board MVP will satisfy:

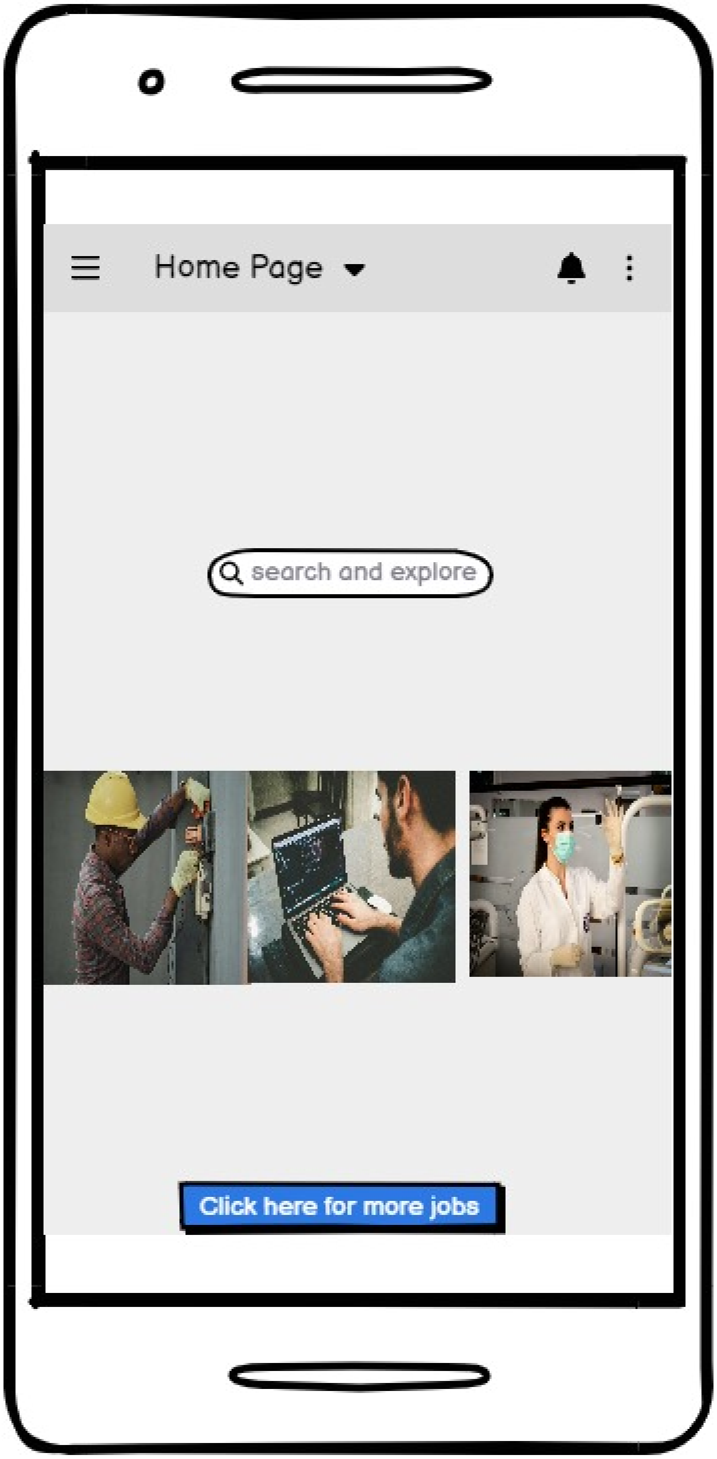
1. As a job seeker, I want to be able to create a profile that includes my personal information, job history, skills, and company details so that employers can find me easily.
2. As an employer, I want to be able to post job listings that include job details such as title, company, location, job description, and application instructions so that job seekers can apply for the job.
3. As a job seeker, I want to be able to search for job listings based on location, job type, and industry so that I can find relevant job openings.
4. As a job seeker, I want to be able to apply for jobs directly through the platform and upload resumes and cover letters so that I can easily apply for jobs.

5. As a user, I want to be able to manage my profiles, job listings, and job applications through a user dashboard so that I can easily keep track of my activity on the platform.

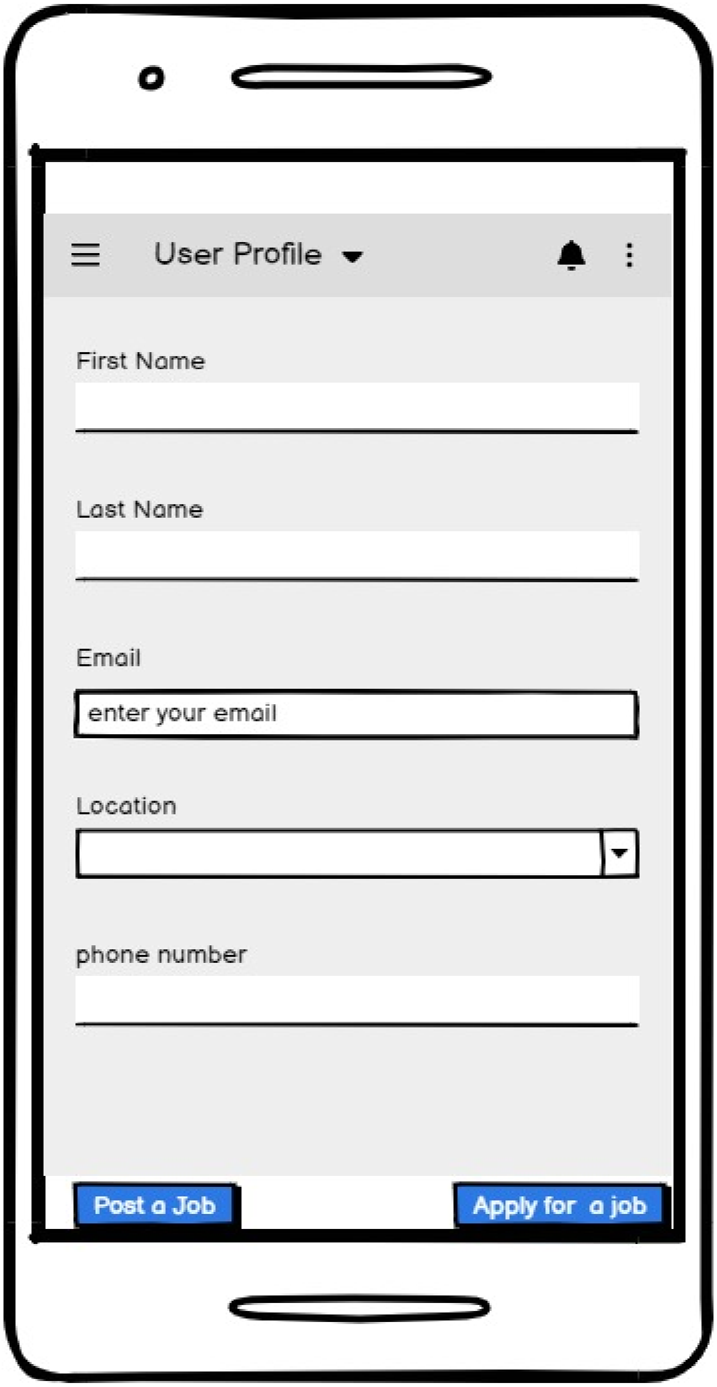
6. As a user, I want to be able to receive email or push notifications about important updates, such as new job listings or application responses so that I can stay up-to-date with the latest information.

Mockups.

Home Page



Mobile Apps


User Profile
First Name
Last Name
Email
enter your email
Location
phone number
Post a Job
Apply for a job