CSE-5335 Web Data Management
Phase – 1 Project Report

Team members:

Achal Bhujangaswami Raghuveer	: 1002102435
Archi Sahu	: 1002104548
Cesar Rea	: 1001643491
Sanmesh Amol Sankhe	: 1001924953
Swaroop Rayapudi	: 1002069356

URM APPLICATION

Introduction:

The Underrepresented Minority Application is a web-based application that offers a platform for URM candidates with the opportunity to apply for Ph.D. studies, postdocs, and faculty positions posted by Academic Institutions, Recruiters, and DEI Officers. This application also facilitates direct communication through a chat feature between URM candidates and Academic Institutions, Recruiters, and DEI Officers.

Academic Institutions, Recruiters, and DEI Officers must register on the website with all the details about their organizations to post jobs in the portal. Similarly, URM candidates must register on the website before applying for any of the jobs.

Overall, the URM application provides a platform for URM candidates to browse various job listings and apply to them.

Entity-Relation Diagram:

An ERD (Entity-relationship Diagram) for a URM application could include the following entities and relationships:

- Users: A person who uses the website.
- Admin: A person who is an admin to the website.
- URM Candidates: A person who wants to apply for the jobs.
- Academic Institution: A person who posts jobs for their institution.
- Recruiter: A person who posts jobs on behalf of their clients/agency
- DEI Officers: A person who posts jobs.
- Application: An entity representing the application submitted by the URM candidates.
- Jobs Posting: An entity representing the Jobs Posted by the academic institution, recruiter, and DEI Officers.

Entities and Relations:

- Users entity will have id, name, email, password and role as attributes where id acts as a primary key. This stores the username and password for all the users.

- Admin entity will have AdminId, email and password where AdminId acts as a primary key. This stores the details of the admin.
- URM Candidates entity will have Uid, name, phone_no, location, nationality, Ethnicity, Education, Res_exp, Publications, Position, Resume and cover letter as attributes where id acts as a primary key of this entity.
- Application entity will have App_id, JId, Uid and status as attributes where App_id acts as a primary key and JId and Uid acts as a foreign key. This entity stores the application details when the candidate applied for the job.
- Recruiters entity will have Jid, Rid, Rname, Agency, Clients and Positions as attributes where Rid acts as a primary key and Jid as a foreign key. This entity stores the details of the recruiters.
- DEI Officers entity will have Jid, Did, Dname, D_Desc, Role, Organization and Position as attributes where Did acts as a primary key and Jid as a foreign key. This entity stores details of the DEI officers.
- Academic institution entity will have Jid, Aid, Aname, A_Desc, Research_focus and positions where Aid acts as a primary key and Jid acts as a foreign key. This entity stores details of the academic institution.
- Faculty entity will have Aid, Fid, Fname, Subject, class and Hrs where Fid acts as a primary key and Aid acts as a foreign key. This entity stores details of the faculty.
- Job postings will have Jid, J_desc, Date, No. of Applicants, Job_position and location where Jid acts as a primary key. This entity stores job details.

Cardinality:

- **URM Candidates - Applications (N:M)**: This relationship will have a many-to-many cardinality as multiple candidates can have multiple applications.
- **Admin - Users (1:M):** This relationship will have a one-to-many cardinality as one admin will send notification to all the users.
- **Users – Users (N:M):** This relationship will have a many-to-many cardinality as multiple users can chat with multiple users.
- **URM Candidates - Job postings(N:M):** This relationship will have a many-to-many cardinality as multiple candidates can apply to multiple jobs.
- **Application – Job postings (N:1):** This relationship will have a many-to-one cardinality as multiple applications belong to a job.
- **Recruiter - Job postings(N:M):** This relationship will have a many-to-many cardinality as multiple recruiters post multiple jobs.
- **DEI Officer - Job postings (N:M):** This relationship will have a many-to-many cardinality as multiple DEI officers post multiple jobs.
- **Academic Institution - Job postings (N:M):** This relationship will have a many-to-many cardinality as multiple academic institutions post multiple jobs.
- **Academic Institution - Faculty (1:M):** This relationship will have a one-to-many cardinality as academic institutions have multiple faculties.

User Roles:

- **Admin:** A person with full access to the application.
- **URM Candidates:** A person who want to apply for the jobs.
- **Academic Institution:** A person who posts jobs for their institution.
- **Recruiters:** A person who posts jobs on behalf of their clients/agency.
- **DEI Officers:** A person who posts jobs.

Technologies to be used: HTML, CSS, NodeJS, ReactJS, PHP, MySQL and Laravel.

Understanding:

First Phase: Will create all the pages using HTML and CSS to enhance the look and feel of the pages.
Second Phase: Then we use React.js a front-end java script library to create a user interface.
Third Phase: In this phase, we will use PHP to store and manage data in MySql DB. It allows us to dynamically access data from the database and incorporate it into the user interface, which results in a more interactive experience for users.
Final Phase: In this phase, we will use a PHP framework called Laravel to integrate all the components and a back-end service called Node.js to enhance real-time processing capabilities.

Through the integration of all these technologies, including the front-end, back-end and database components, we will create a comprehensive and dynamic URM application.

In terms of a comprehensive viewpoint, a URM application can have users as URM candidates, Academic Institutions, Recruiters, DEI officers, and admin. Admin has full access to the application. They can perform CURD operations for all other users. They can view all the jobs posted by the users. Also, they can view all the applications with their status submitted for jobs. They have access to generate different reports such as the total number of applications submitted for available jobs, show the total number of jobs posted by different users, and report about URM candidates who are selected for the jobs. They can send the notification to the candidates who are selected. URM candidates have access to update their profile details after registration. They can search/apply for jobs and view their application status. They can chat with academic institutions, recruiters, and DEI officers for their questions. Recruiters have access to manage/update their profile such as details about their clients, agency, and positions they are recruiting for. They can post/view the jobs, view the applications received with the status, and view candidates matching their preferences. They have the option to chat with candidates, DEI officers and academic institutions. Similarly, Academic Institutions can manage/update their profile, and they can add their faculty details. They can post/view the jobs, view the applications received with the status, and view candidates matching their preferences. They have the option to chat with candidates, DEI officers and recruiters. DEI Officers can manage/update their profile. They can post/view the jobs, view the applications received with the status, and view candidates matching their preferences. They have the option to chat with candidates, recruiters, and academic institutions.