CS 387 Project: NetWorks

Pandurang Deore (200050096)
Sarthak Mittal (200050129)
Shikhar Agrawal (200070076)
Mayank Jain (20d070050)

Contents

1 Summary

1.1 Description

We plan to build a clone of the LinkedIn Web application. We intend to incorporate most features of an online job hub. We would be implementing the same in a PERN stack framework (PostgreSQL, Express, React, Node.js).

1.2 Flow of the System

Navigating through the website, we will create a signup/login interface, user dashboard (with feed), and profile page, and maintain data of connections among people. There will be two types of users, recruiters and applicants. We will add some data by default for the demonstration and will accept new data (profiles, posts, job postings, connection requests) from logged-in users. If time permits, we will also create a chat interface in the app.
For the details of the components, we plan to have multiple fields while creating a profile, the option to add media to posts, and details of job postings (recruiter, company, location, etc.). If time permits, we will also try to visualize the connections using a graph.

2 Overview

2.1 Domain

This application will be part of online professional networks and employment portals. We intend to incorporate most features of an online job hub.

2.2 Intended Use

The application will support:

- building a portfolio and connecting with people of similar interests
- staying up to date with the professional network and opportunities
- broadcasting, searching and applying for jobs online
- real-time chatting with others in your professional network[1]

3 Users

There would be broadly two types of users: recruiters and applicants.

[1] if time permits

3.1 Role of a Recruiter

- looking for applicants to hire for the job postings
- gives the posting details such as company, location, duration, etc.
- waits for applicants to apply and then contacts them accordingly

3.2 Role of an Applicant

- looking for job postings to apply to and applies accordingly
- gives the profile details such as name, email, qualification, etc.
- waits for recruiters to accept their application

4 Database

There would be three entities - users, posts and jobs.

4.1 User Entity

- userID, password, name, address, contact, typeOfUser
- qualifications and experience
- connections
- posts
- jobs (applied for/posted)

4.2 Post Entity

- postID, userID
- content
- timeOfPost
- reactions
- comments[2]

[2] if time permits

4.3 Job Entity

- jobID, userID
- company, location, duration, etc.
- applicants

5 Application

5.1 Backend API

We will provide endpoints for the following functionality:

- login/signup
- feed
- profile
- connections
- job board
- chat[3]

5.2 User Interface

The web application will provide interfaces for the following functionality:

- login/signup
- navigation bar
- dashboard
- profile
- network
- jobs
- chats[3]

5.3 Security

We will use Express for session management.

[3] if time permits