

Team Member Full Name	NetID
Alex Wasinger	jmunhall
Jack Munhall	mbai
Michael Bai	awasing2
Brian Scherer	bschere2

Chosen Technology Stack

Python + Django

Python + TkInter

Persistent Storage Design

We are using SQLite database to persist our data. Our database includes the tables shown in Figure 1. It contains 5 tables: User, Recruiter, Candidate, Post, and Offer, to represent each of the classes in our database. The first table is the User table, which inherits from Django's built-in AbstractUser class. The Candidate and Recruiter classes have User fields, which represent a created Django "OneToOneField," which connects these classes to the User class. Additionally, the Post and Candidate fields in the Offer class, the Interested_Posts and Not_Interested_Posts fields in the Candidate class, and the Recruiter field in the Post class each represent an instance of another class (and are connected to the appropriate classes in the diagram) – therefore their type is "int?" because the software used in the diagram doesn't recognize the types of those fields (because the types are Class names).

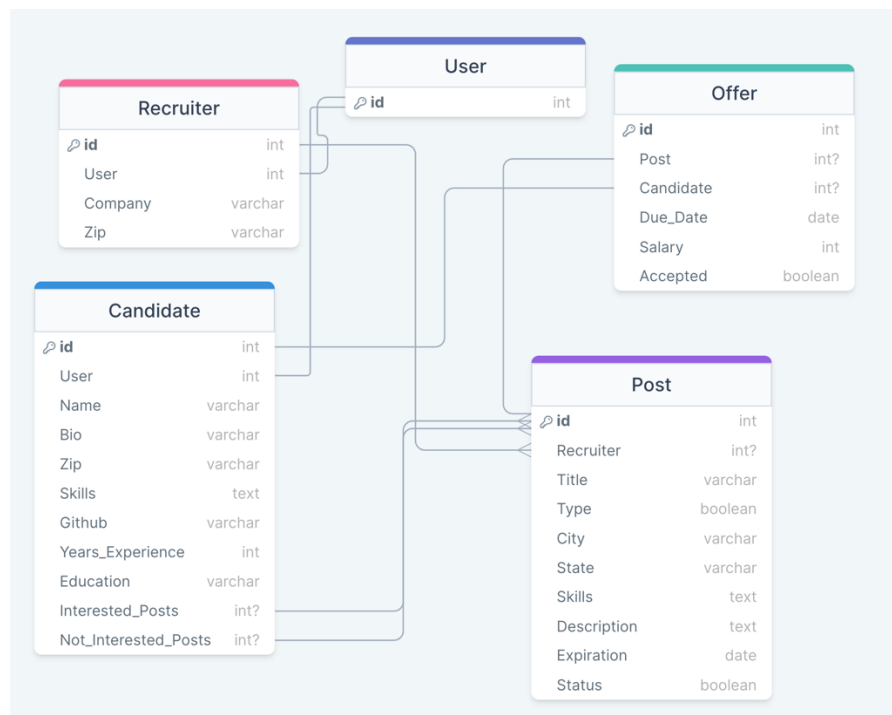


Figure 1 database schema

Demonstration of the Features

Feature 1.1

Figure 1.1A shows a screenshot for Feature 1.1: Create a new Candidate Profile. It shows the registration page, with information entered. Figure 1.1B shows the candidate dashboard displayed when the registration is successfully submitted with the “Submit” button, and the candidate has been created.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

FAZZ2- X CSE- X Lactur X FAZZ2 P X parad X reasing X Studer X Home X Your C X Phase X Top 5 X TrnDe X 129.74 X FAZZ2- X Linux- X

Not Secure | 129.74.152.125:51095/register

Here, you can register for our platform as a candidate or as a recruiter.

Create Candidate

Username
User11

Password

Name
Patrick

Bio (Max 500 char.)
This is Patrick

Zip Code
46556

Skills
Linux, Python, Java

GitHub
user11

Years of Experience
6

Education
Notre Dame

Figure 1.1A: Screenshot for feature 1.1 showing the candidate registration page

Chrome File Edit View History Bookmarks Profiles Tab Window Help

FAZZ2- X CSE- X Lactur X FAZZ2 P X parad X reasing X Studer X Home X Your C X Phase X Top 5 X TrnDe X 129.74 X FAZZ2- X Linux- X

Not Secure | 129.74.152.125:51095/candidate_dashboard

TinDev Dashboard Logout

Welcome Candidate!

Here you can view all jobs posted by recruiters as well as any potential offers.

In addition, you can [favorite](#) posts.

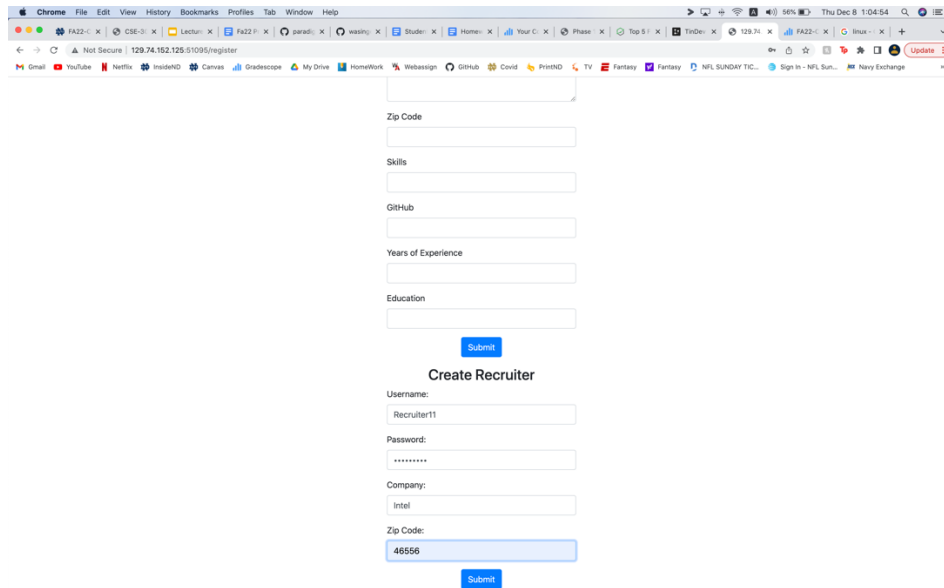
[Click here to view all job postings](#)

My Favorites Job Offers

Figure 1.1B: Screenshot for feature 1.1 showing the candidate dashboard

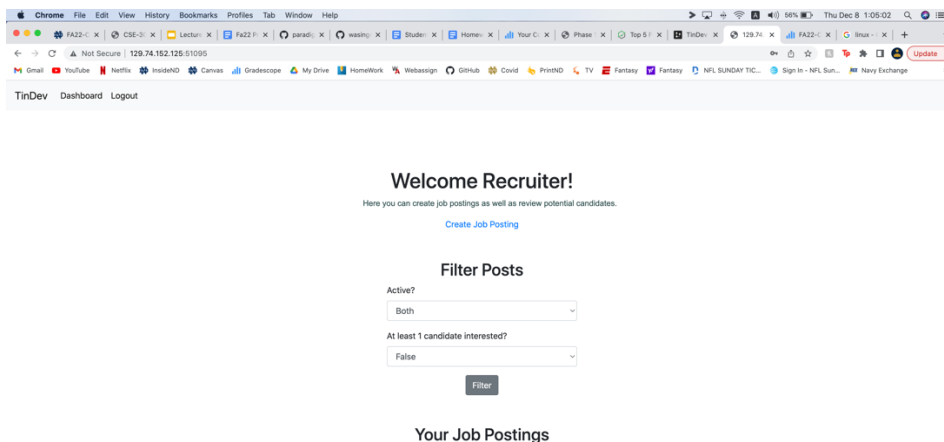
Feature 1.2

Figure 1.2A shows a screenshot for Feature 1.2: Create a new Recruiter Profile. It shows the registration page, with information entered. Figure 1.2B shows the recruiter dashboard displayed when the registration is successfully submitted with the “Submit” button, and the recruiter has been created.



A screenshot of a web browser showing a registration form for a recruiter. The form is titled "Create Recruiter" and contains several input fields: "Zip Code", "Skills", "GitHub", "Years of Experience", "Education", "Username" (with "Recruiter11" entered), "Password" (with "*****" entered), "Company" (with "Intel" entered), and "Zip Code" (with "46556" entered). There are two "Submit" buttons, one above the "Education" field and one below the "Zip Code" field. The browser's address bar shows "129.74.152.125:51095/register".

Figure 1.2A screenshot for feature 1.2 showing the recruiter registration page



A screenshot of a web browser showing the recruiter dashboard. The page has a header with "TinDev", "Dashboard", and "Logout" links. The main content area is titled "Welcome Recruiter!" and includes a sub-header "Here you can create job postings as well as review potential candidates." with a "Create Job Posting" link. Below this is a "Filter Posts" section with two dropdown menus: "Active?" (set to "Both") and "At least 1 candidate interested?" (set to "False"). A "Filter" button is located below these dropdowns. The section "Your Job Postings" is visible at the bottom of the page.

Figure 1.2 screenshot for feature 1.2 showing the recruiter dashboard

Feature 1.3

Figure 1.3 shows a screenshot for Feature 1.3: Log-in. It shows information entered in from a previously created candidate account.

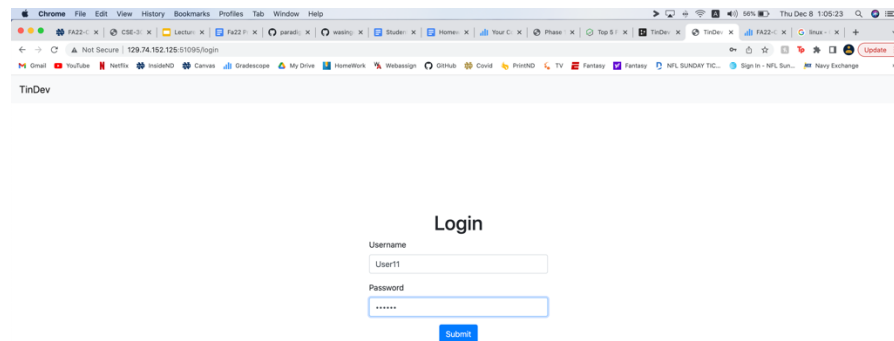


Figure 1.3 screenshot for feature 1.3 showing the login page

Feature 1.4

Figure 1.4A shows a screenshot for Feature 1.4: Log-out. It shows the candidate dashboard of the website with the logout button. Figure 1.4B shows the empty log-in page, where the log-out button takes the user.

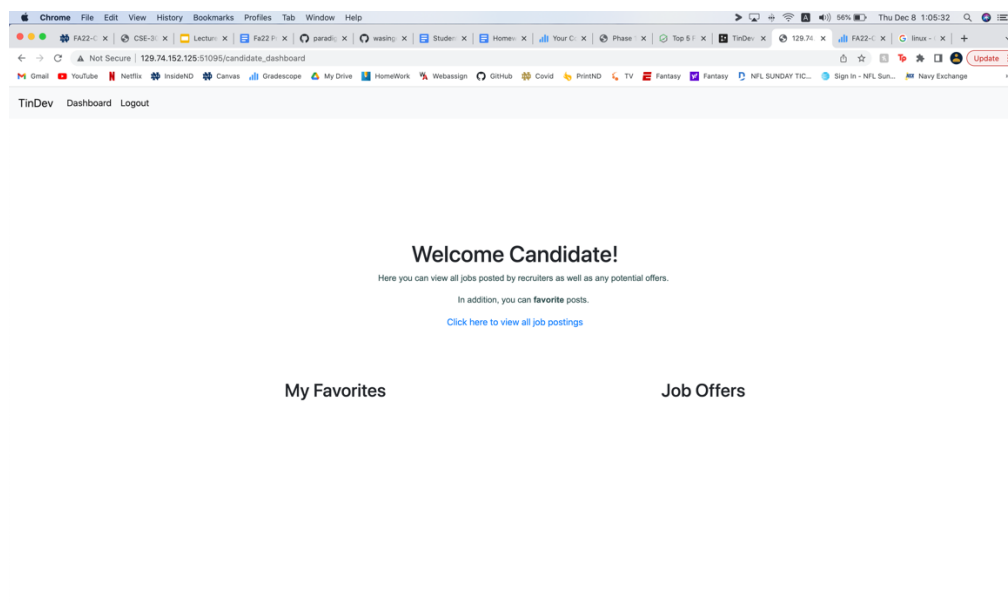


Figure 1.4A screenshot for feature 1.4 showing the candidate dashboard

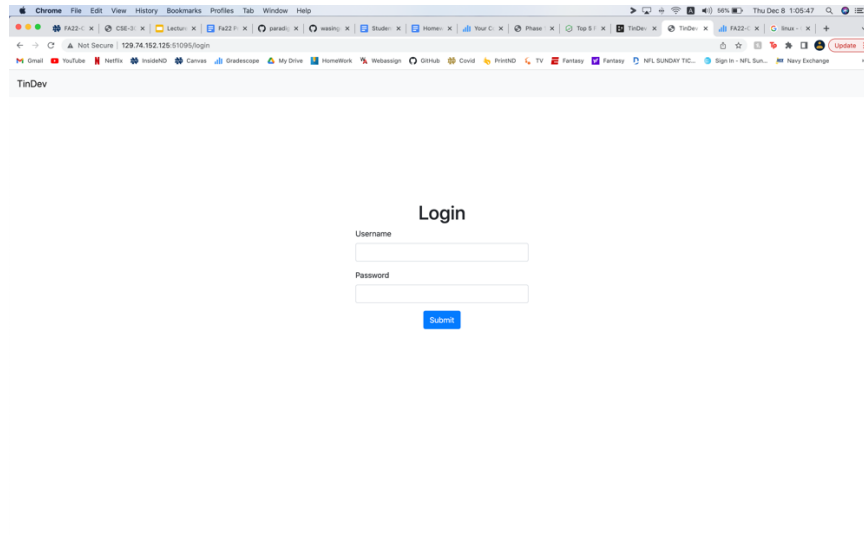


Figure 1.4B screenshot for feature 1.4 showing successful log-out

Feature 2.1

Figure 2.1 shows a screenshot for Feature 2.1: View all posts. It shows the recruiter all posts they have posted.

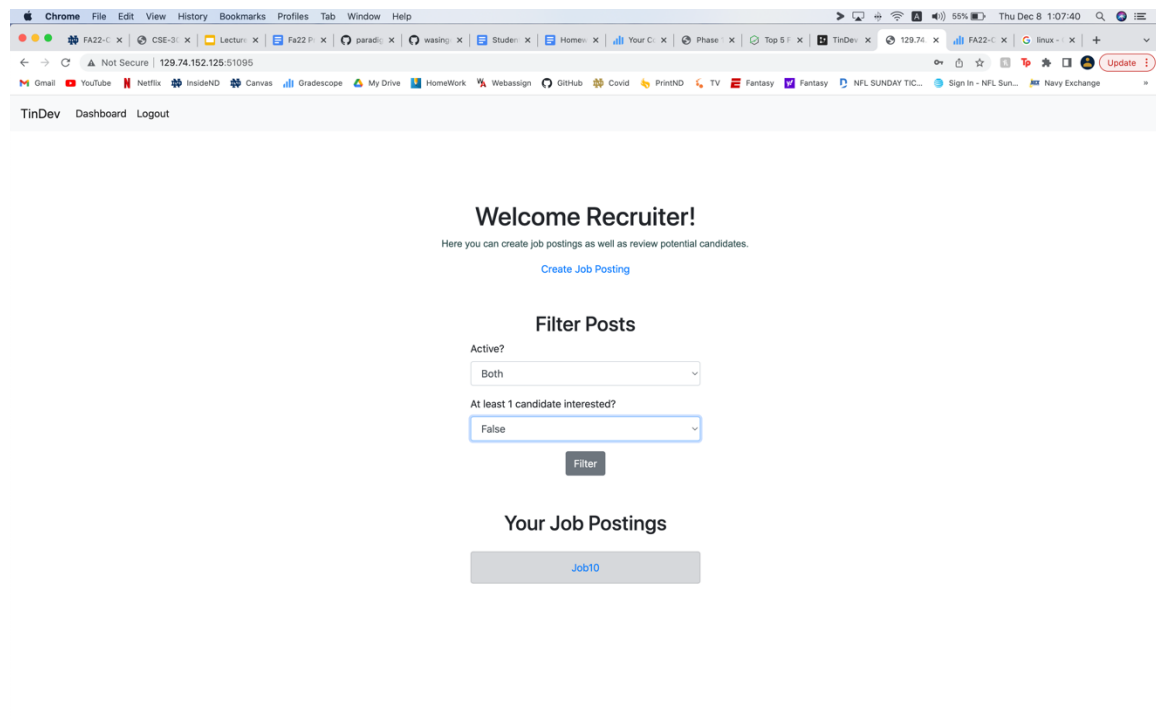


Figure 2.1 screenshot for feature 2.1 showing the recruiter's job posts

Feature 2.2

Figure 2.2A shows a screenshot for Feature 2.2: Creating a post. It shows the post creating page for the recruiter. Type indicates full-time (checked) or part-time (unchecked). Status indicates whether the post is active (checked) or inactive (unchecked). Figure 2.2B shows the successfully created job post.

Create Job Posting

Title:
Job12

Type: ☒ Full-time ☐ Part-time

City:
Chicago

State:
Illinois

Skills (comma-separated):
Python, Github, HTML, Java

Description:
This is Job12

Expiration (MM/DD/YYYY):
04/06/2023

Status: ☐ Active ☒ Inactive

[Submit](#)

Figure 2.2A screenshot for feature 2.2 showing the recruiter's post creation page

Welcome Recruiter!

Here you can create job postings as well as review potential candidates.

[Create Job Posting](#)

Filter Posts

Active?
Both

At least 1 candidate interested?
False

[Filter](#)

Your Job Postings

[Job10](#)

[Job12](#)

Figure 2.2B screenshot for feature 2.2 showing the created post

Feature 2.3

Figure 2.3A shows a screenshot for Feature 2.3: Updating a post. It shows the page which allows recruiters to update job posts. Figure 2.3B shows the details page for the successfully updated job post.

The screenshot shows a web browser window with the URL `129.74.152.125:51095/posts/update/14`. The page title is "Update Job Posting". The form contains the following fields:

- Title:
- Type: ☐
- City:
- State:
- Skills (comma-separated):
- Description:

This is Job10, UPDATED
- Expiration (MM/DD/YYYY):
- Status: ☐

At the bottom of the form is a "Submit" button. Below the form is a section titled "Delete Job Posting" with a "Delete" button.

Figure 2.3A screenshot for feature 2.3 showing the page for updating a job post

The screenshot shows a web browser window with the URL `129.74.152.125:51095/posts/14`. The page title is "Job Details". The page displays the following information:

Position Title: Job10
Full Time or Part Time? False
City: Chicago
State: Illinois
List of Preferred Skills: Linux, Github, Python, Java, JavaScript, HTML
Job Description: This is Job10, UPDATED
Job Expiration Date: July 3, 2023
Status (Active / Inactive): False

Below the table is a section titled "Interested Candidates". It contains the following fields:

- Starting Salary (USD):
- Reply to Offer By:

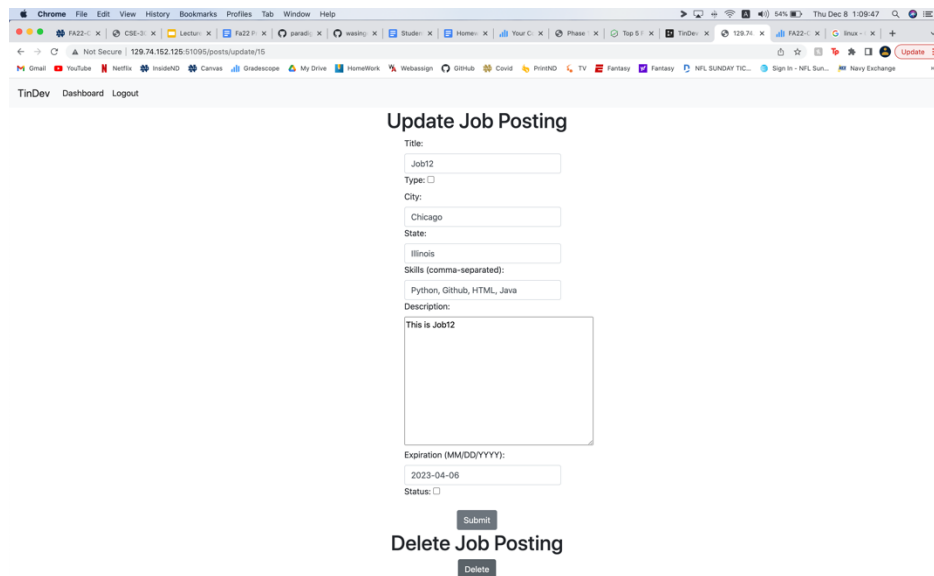
At the bottom of the "Interested Candidates" section is a "Send Offer" button.

Below the "Interested Candidates" section is a section titled "List of Offers Sent".

Figure 2.3B screenshot for feature 2.3 showing the successfully updated job post page

Feature 2.4

Figure 2.4A shows a screenshot for Feature 2.4: Deleting a post. It shows the delete button, located at the bottom of the update page. Figure 2.4B shows the updated job posts list, with the deleted job now missing as is desired.

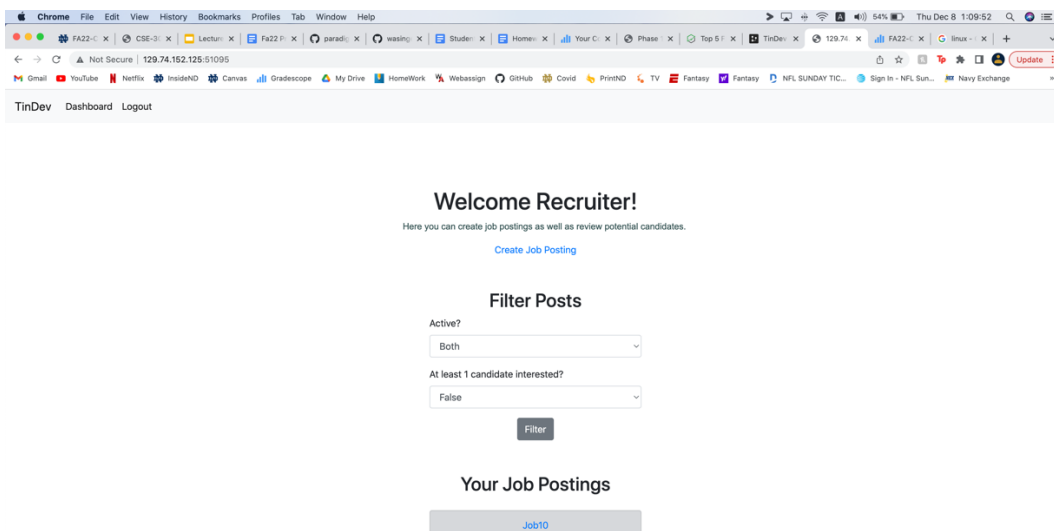


The screenshot shows a web browser window with the URL `129.74.152.125:51095/posts/update/16`. The page title is "Update Job Posting". The form contains the following fields:

- Title:
- Type: ☐
- City:
- State:
- Skills (comma-separated):
- Description:
- Expiration (MM/DD/YYYY):
- Status: ☐

At the bottom of the form, there is a "Submit" button and a "Delete" button. The "Delete" button is highlighted with a red circle.

Figure 2.4A screenshot for feature 2.4 showing the delete button at the bottom



The screenshot shows a web browser window with the URL `129.74.152.125:51095`. The page title is "Welcome Recruiter!". The page content includes:

- A message: "Here you can create job postings as well as review potential candidates." with a link "Create Job Posting".
- A section titled "Filter Posts" with two dropdown menus: "Active?" (set to "Both") and "At least 1 candidate interested?" (set to "False"). Below these is a "Filter" button.
- A section titled "Your Job Postings" with a list of job postings. The only visible posting is "Job10".

Figure 2.4B screenshot for feature 2.4 showing the lack of the successfully deleted post

Feature 2.5

Figure 2.5A shows a screenshot for Feature 2.5: Make an offer to interested candidates. It shows the offer fields (candidates selected, starting salary, offer expiration) filled. Figure 2.5B shows the list of the successfully made offers.

City: Chicago

State: Illinois

List of Preferred Skills: Linux, Github, Python, Java, JavaScript, HTML

Job Description: This is Job10, UPDATED

Job Expiration Date: July 3, 2023

Status (Active / Inactive): False

Interested Candidates

Adam space 50.0% match ☒

Stephen space 16.666666666666664% match ☐

Michael space 16.666666666666664% match ☐

Elizabeth space 83.33333333333334% match ☒

Starting Salary (USD):

70000

Reply to Offer By:

01/17/2023

Send Offer

List of Offers Sent

Update Post

Update

Figure 2.5 screenshot for feature 2.5 showing the offer making functionality

State: Illinois

List of Preferred Skills: Linux, Github, Python, Java, JavaScript, HTML

Job Description: This is Job10, UPDATED

Job Expiration Date: July 3, 2023

Status (Active / Inactive): False

Interested Candidates

Adam space 50.0% match ☐

Stephen space 16.666666666666664% match ☐

Michael space 16.666666666666664% match ☐

Elizabeth space 83.33333333333334% match ☐

Starting Salary (USD):

mm/dd/yyyy

Reply to Offer By:

mm/dd/yyyy

Send Offer

List of Offers Sent

Adam	Jan. 17, 2023
Elizabeth	Jan. 17, 2023

Update Post

Update

Figure 2.5B screenshot for feature 2.5 showing the successfully made offers

Feature 2.6

Figure E shows a screenshot for Feature 2.6: View compatibility with interested candidates. It shows the candidates on a job posting, each followed by their compatibility with the post. The compatibility score is calculated by finding the percent of skills desired by the job post which each candidate has.

The screenshot shows a web browser window with a job posting form and a list of interested candidates. The job posting details include: State: Illinois, List of Preferred Skills: Linux, Github, Python, Java, JavaScript, HTML, Job Description: This is Job10, UPDATED, Job Expiration Date: July 3, 2023, and Status (Active / Inactive): False. Below this, the 'Interested Candidates' section lists four candidates with their compatibility scores: Adam space 50.0% match, Stephen space 16.666666666666664% match, Michael space 16.666666666666664% match, and Elizabeth space 83.33333333333334% match. There is a 'Starting Salary (USD):' field and a 'Reply to Offer By:' field with a date picker. A 'Send Offer' button is present. Below the candidates, the 'List of Offers Sent' section shows offers sent by Adam on Jan. 17, 2023, and Elizabeth on Jan. 17, 2023. At the bottom, there is an 'Update Post' button with a blue 'Update' link below it.

Figure 2.6 screenshot for feature 2.6 showing the compatibility of posts and candidates

Feature 3.1

Figure 3.1 shows a screenshot for Feature 3.1: Viewing job postings. It shows the candidate's view of the job postings from all recruiters.

The screenshot shows a web browser window displaying a list of job postings. The title is 'Job Postings' with a subtitle 'The following are job postings made by a recruiter:'. Below this is a 'Filter Posts' section with a 'Currently active?' dropdown menu set to 'Both', and fields for 'Location' and 'Keywords'. A 'Filter' button is located below these fields. The list of job postings is displayed as a series of buttons: 'First post', '1', '2', '3', 'Job1', 'Job2', and 'Job3'.

Figure 3.1 screenshot for feature 3.1 showing all job postings

Feature 3.2

Figure 3.2A shows a screenshot for Feature 3.2: Demonstrating interest / not interested in a job. It shows the checkbox for submitting application (demonstrating interest). If this box were unchecked, then no application would be sent (and the candidate is not interested in the job). Figure 3.2B shows the job added to the candidate's favorites (if the submit application box had not been checked it would not be there).

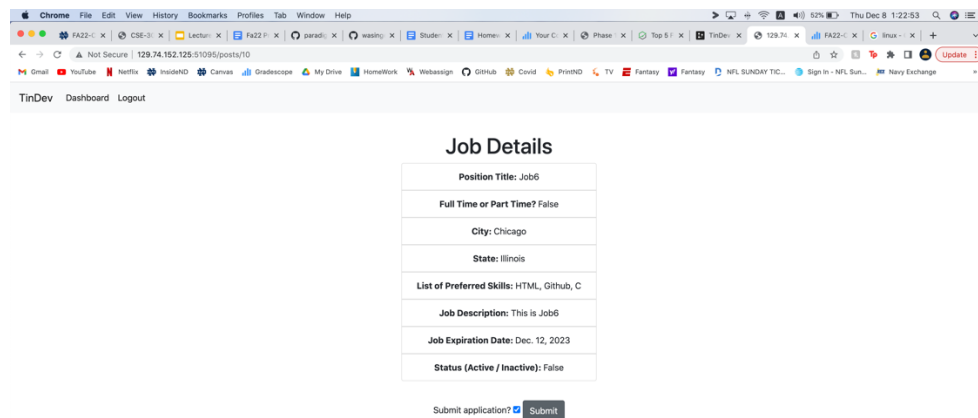


Figure 3.2 screenshot for feature 3.2 showing the checked box in the job post

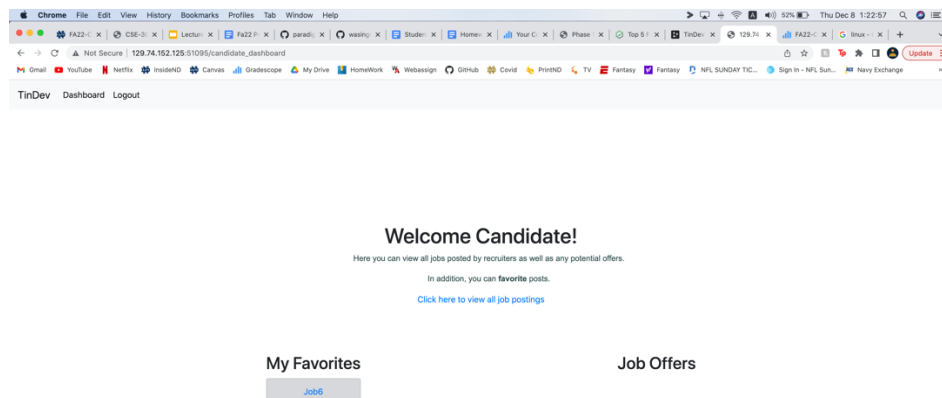


Figure 3.2B screenshot for feature 3.2 showing the job added to the candidate's favorites

Feature 3.3

Figure E shows a screenshot for Feature 3.3: Viewing offers. It shows the candidate dashboard with a listed job offer for this candidate.

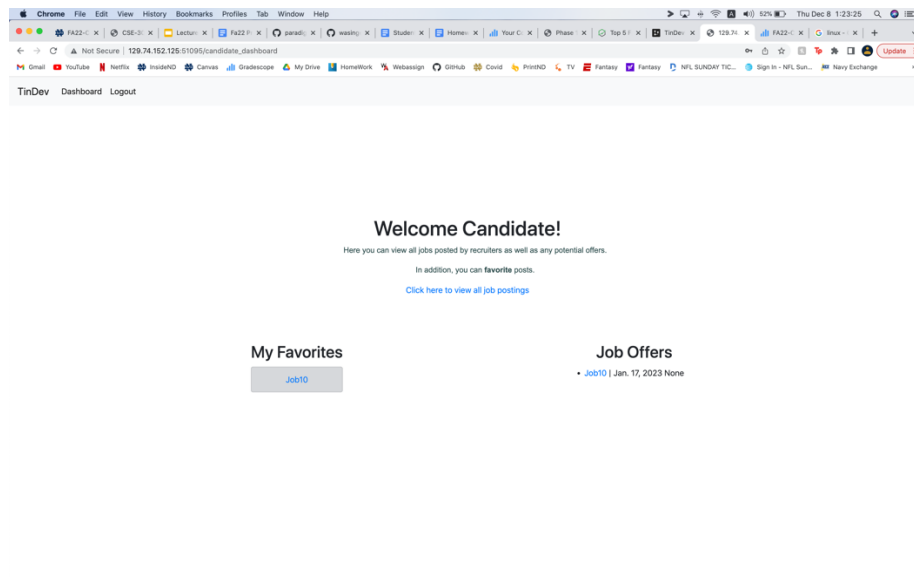


Figure 3.3 screenshot for feature 3.3 showing the job offers for this candidate

Feature 3.4

Figure 3.4A shows a screenshot for Feature 3.4: Accepting/declining offers. It shows the offer page and the candidate's option to accept (checked) or decline (unchecked). Figure 3.4B shows the candidate dashboard with the newly accepted and declined offers for a candidate.

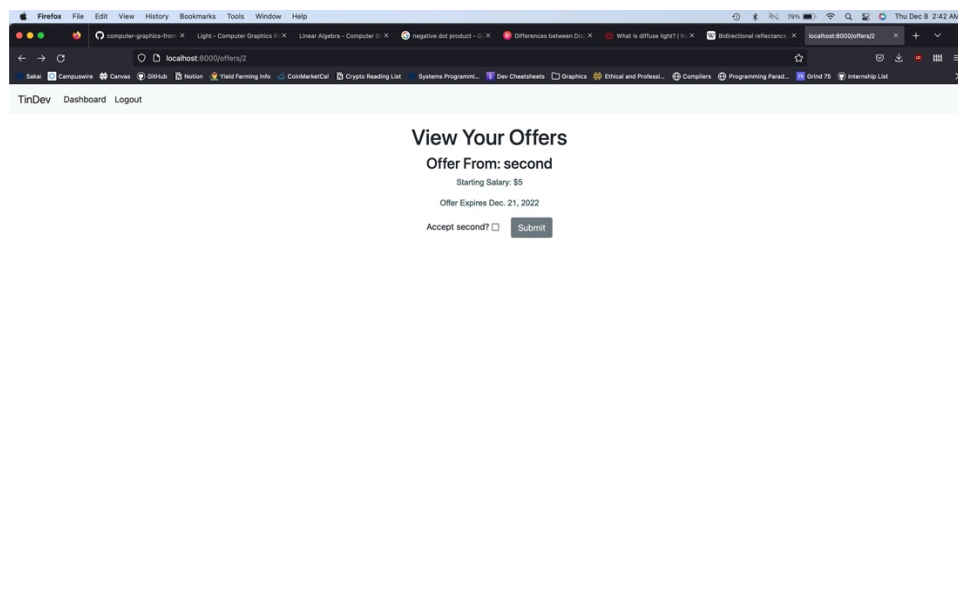


Figure 3.4A screenshot for feature 3.4 showing the offer page

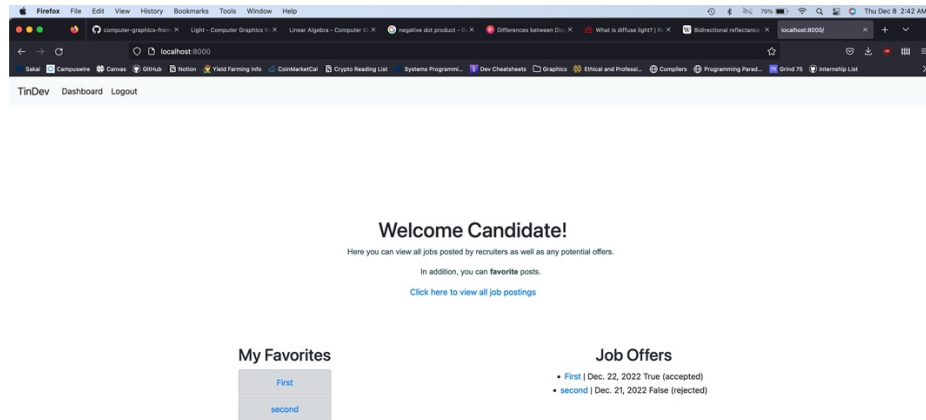


Figure 3.4B screenshot for feature 3.4 showing the candidate dashboard with new decisions on the offers

Project's Learned Lessons

1. *What programming paradigm(s) have you chosen to use and why? If you were to start the project from scratch now, would you make different choices? Do you think the paradigm(s) chosen helped (or not) in developing the project?*

We used imperative programming by using python for all of the back end functions and classes. In addition, we used predominantly object-oriented programming on both the front and back ends. On the back end, this involved using Django's MVT (Model-View-Template) infrastructure, using objects such as Posts, Recruiters, and Candidates to organize the functionality of our project. On the front end, we utilized OOP by employing template inheritance, which made it easy to create all the different pages on the website.

2. *What were the most intellectually challenging aspects of the project?*

This project gave us good practice in reading and comprehending online documentation, as we had to understand the Django pages online. Additionally, we had good experience using GitHub to collaborate as a team. Finally, it was a good challenge to practice thinking about the design considerations involved with the appearance of the website.

3. *What aspects of your process or your group's organization had the largest positive effect on the project's outcome?*

Our group excelled at covering different parts of the requirements of this project individually and then bringing everything together. There was a clear differentiation between those working on the front end and those on the back end.

This allowed everyone to be productive on their own and best use their strengths, while still allowing us to consolidate everything at the end.

Overall, our group worked well in completing this project, and we are happy with how it turned out and the experience and knowledge we gained.