

FULLSTACK DEVELOPER FINAL ASSIGNMENT

Functional and technical design

Attemp-2

Student name: Mustafa Dogan

2/10/2025

Figma project link: <https://www.figma.com/community/file/1478846842989482623>

Github Project link: <https://github.com/ddOgan/vacancy/>

Vacancy Web is a Java spring-boot react web application that allows the system owner to post the vacancy and let people apply for the position.

Table of contents

Table of contents	2
List of diagrams.....	4
List of Table.....	6
1 Introduction.....	7
1.1 Problem Description	7
1.2 Scope.....	8
1.3 Inspiration.....	8
2 Functional design specification	9
2.1 Project requirements	9
2.1.1 Functional requirements.....	10
2.1.2 Non-functional requirements.....	12
2.2 Application Process Flow.....	13
2.2.1 User Signup/Sign-In flow	13
2.2.2 Job vacancies process flow.....	14
2.2.3 User profile management flow.....	15
2.3 Use case.....	16
2.3.1 Use case diagram	16
2.3.2 Actors	16
2.3.3 Use case table	17
2.4 Wireframes/Mockup screens	21
2.4.1 Employer sign-up	23
2.4.2 Employer sign-in	24
2.4.3 Employer Post new Job openings	24

2.4.4	Employer view Job openings	25
2.4.5	View Applicants	26
2.4.6	Employer notifications	26
2.4.7	Candidate Sign-Up	27
2.4.8	Candidate Sign-In	28
2.4.9	Candidate search/view Jobs	28
2.4.10	Candidate apply for the job opening	29
2.4.11	Candidate view notifications	30
3	Technical design specification	31
3.1	Technical stack.....	31
3.1.1	Back-end.....	31
3.1.2	Front-end.....	31
3.1.3	Database.....	32
3.2	System UML design.....	32
3.2.1	Front-end.....	32
3.2.2	Back-end.....	33
3.3	API design	33
3.3.1	Auth api (/api/auth).....	33
3.3.2	User api (/api/users)	34
3.3.3	Vacancy api (/api/vacancies).....	35
3.4	Security configuration	36
3.5	Sequence diagram	38
3.5.1	User sign-up/ sign-in	38
3.5.2	Vacancy flow.....	39
3.5.3	User profile flow	41
3.6	Class diagram	41

Vacancy Web application	Functional and technical design
4 Window design	41
4.1 Candidate Sign-Up	42
4.2 Candidate Sign-In	43
4.3 Candidate dashboard.....	43
4.4 Candidate Profile	44
4.5 Candidate view job openings	46
4.6 Candidate Notifications.....	47
4.7 Candidate settings	47
4.8 Employer user Signup	48
4.9 Employer user Sign-in	48
4.10 Employer dashboard	49
4.11 Employer post new vacancy	49
4.12 Employer Job posts	50
4.13 Employer Review candidate requests.....	51
4.14 Employer Notifications.....	52
4.15 Employer settings	53
5 Conclusion	53
References	54

List of diagrams

Figure 1: Process flow User Sign-up and Sign -in.....	13
Figure 2: Process flow Job vacancies	14
Figure 3: Process flow user profile management.....	16
Figure 4: Use-case diagram	16
Figure 5: wireframe employer Sign-Up	23
Figure 6: wireframe employer Sign-In	24

Figure 7: wireframe employer post new job.....	25
Figure 8: wireframe employer view job openings	25
Figure 9: Employer view candidate requests	26
Figure 10: wire frame employer notifications.....	27
Figure 11: wireframe candidate Sign-up	27
Figure 12: wireframe candidate sign-in	28
Figure 13: wireframe view/search job opening.....	29
Figure 14: wireframe candidate apply for job opening	30
Figure 15: wireframe candidate notifications	31
Figure 16: Frontend UML diagram.....	32
Figure 17: Backend UML Diagram.....	33
Figure 18: Security configuration setup	37
Figure 19: JWT Token filter.....	38
Figure 20: Sequence diagram (User sign-in or sing-up)	39
Figure 21: Sequence diagram (View vacancies and apply for job openings)	40
Figure 22: Class diagram.....	41
Figure 30: Candidate Sign-up window	42
Figure 31: Candidate Sign-in window	43
Figure 32: Candidate dashboard.....	43
Figure 33: window t upload candidate CV	45
Figure 34: Window to apply job opening	46
Figure 35: Candidate notifications.....	47
Figure 36: Candidate save settings	48
Figure 37: Employer registration window	48
Figure 38: Employer sign-in window.....	48
Figure 39: Employer dashboard.....	49
Figure 40: window to post new job	50
Figure 41: employer view jobs.....	50
Figure 42: window to review candidate applications	52
Figure 43: Employer notification window	52
Figure 44: Employer settings	53

List of Table

Table 1: System actors	17
Table 2: Use case table 1 Employer/organization Sign-Up	18
Table 3: Use case table 2 Employer/organization Sign-In.....	18
Table 4: Use case table 3 Employer post new job opening	19
Table 5 Use case table 4-Employer view applicants for job.....	20
Table 6 Use case table 5-Candidate search jobs	21
Table 7 Use case table 6 Candidate apply for job openings.....	21

1 Introduction

Today vacancy web apps are one of the essential online tools where employees can register, post and search jobs (openings). It basically helps job seekers to find and apply for relevant jobs. Vacancy based web applications can provide an easy platform to search available job positions based on interest and skills, enabling them to submit their resume directly through our website. Here I have developed a vacancy Web application using Java spring-boot react.

1.1 Problem Description

Conduct recruitment process is very difficult without automated tools or software application; it has many problems such as

- **Time-Consuming Processes**

It requires time manually collecting resumes and screening resumes according to skill set. Sometimes time is a tedious process to maintain applicant records on paper.

- **Human errors**

The person who collects details can have errors in data entry, tracking application or interview scheduling, misplace or loss resume.

- **Struggling**

Employers require more struggles to maintain an application record for multiple jobs when there is high volume resumes especially when multiple positions simultaneously becomes overwhelming.

- **Administrative burden**

Recruiter required spending extra time to sort resume, filter papers and select most skilled employees from the list.

- **Data management problems**

Manually information tracking increases risk of information loss as well as problems with reusability.

- **Geographical limitations**

It is biggest problems, only limited employee cans rich to the employer place. It is not possible to select skilled employees from various locations.

Here our vacancy web application can solve those problems and provide an easy platform for candidates as employers. Online job portals will be very helpful to them because it can give a centralized platform for employees to access a variety of job opportunities from multiple companies, and it also makes it easy for employers to search highly skilled employees for job openings. It is streamlining the recruitment process for both employee and employer. It will be useful to employers to search for a skilled candidate, company profile, post jobs, maintain and store resumes, and other candidates can also easily register with the portal, search job openings, apply to their relevant jobs, update resume and manage profile.

1.2 Scope

Business users are looking to build a web platform that allows users to create and manage the job advertisements and job applicants; by having that integrated platform, business users could easily track the number of jobs and applicants that existed in the system and allow them to plan ahead the job advertising schedule.

The project is aiming to deliver the first interim solution that provides the job posting and job application functionalities along with user registration and user profile management.

1.3 Inspiration

I was inspired through a web platform (i.e. cv-library, Indeed and the Guardian Jobs) that provides seamless facilities to organization and candidates. They provide multiple features with attractive front-end layout which supports mobile, tablet as well as desktop screen without compromising website layout.

Another I found that developing a vacancy web application using java spring boot requires a blend of functional efficiency as well as user-friendly aesthetics. Further below some visual inspiration to guide me to design our project.

1. Vacancy web application design using CMARIX. This helps to develop a harmonious blend of different color, space and create Interfaces. It is inspired to design the main

- essentials for job seekers such as layout design, highlights clarity, ease of navigation etc... (dribbble, 2025)
2. Pinterest has a huge collection of web pages offering clean and modern aesthetics, mainly this focus on creating user involvement through proper navigation and prominent action buttons. This helps to enhance organization of website control and sections for more readability. (pinterest, 2025)
 3. There are a number of paid job portal designs which inspire us to make desired changes on our web portal such as the Jobie Admin panel. This provides a guide to make comprehensive interface design with well-structured data representation. This helps to balance functionalities with ensuring user efficiency and admin can manage job posts. (DexignZone, 2025)

Here I also tried to design web application by considering best practices:

1. **Accessibility:** include app features that develop websites more usable for every individual with disabilities.
2. **User centric design:** that ensure user interface develop intuitive, enable user to navigate effortlessly
3. **Performance optimization:** Here I also tried to optimize assets (i.e. logo, pictures, code, library) to ensure quick load time. This enhances user satisfaction level.
4. **Responsive Layout:** Design User interface in a way that displays consistent across multiple screen sizes and devices.

Further found both Java Spring Boot (i.e. backend technology) and React (i.e. frontend technology) will enhance efficiency of our job portal web application.

2 Functional design specification

2.1 Project requirements

This section basically describes a list of software functionalities and features separated in two categories 1) Functional requirements and 2) non-functional requirements. Both types of requirements are useful for successful software development.

2.1.1 Functional requirements

Functional requirements describe what the system actually does. Mainly this includes software features, tasks, and communication windows.

2.1.1.1 Employer user requirements

1. Need interface to register employers and create their account using company name, username and password in our web application.
2. Need an employer authentication window to enter into their account.
3. Required an interface to design update company profile includes name, description, logo, contact number, email address, and office address.
4. Requires to create employer dashboard to view account summary (i.e. number of vacancy posted, number of applicants, and number of notifications)
5. Require an interface to post new job openings by entering job title, description, required experience, skill, no of seats, application due date, position, and job location.
6. Provide a window to view a list of candidates required for a particular posted job. It needs to sort and filter job applications through experience level, expected salary package, and skill.
7. Provide a window to view candidate profiles that include personal information, professional information, experience, skill as well as resume.
8. Need a notification if any candidate applied for their job.
9. Employers should be able to update the posted vacancy status like Cancel.
10. Employers should be able to select candidates from a posted job and send them notification for the final interview or call later.
11. Provide an interface to view candidate profiles.
12. Employers should be able to update the posted vacancy status like Cancel.
13. Need an interface to review candidate application requests, in particular posted jobs.
14. Need employer account setting window to configure employer account (such as change password, delete account, etc....)
15. Need sign-out button to exit and terminate employer user session parameters.

2.1.1.2 Candidate user requirements

1. Need an interface to register candidates and create their account using username and password in our web application.
2. Require authentication window where candidates can enter into their account.
3. Provide a search interface to find job openings. (also need filter job based on location, keyword and skills)
4. Provide interface to view employer profile.
5. Develop a job detailing interface to view all the information about job openings.
6. Requires to create candidate dashboard to view account summary (i.e. total of applied, selected notification (if any), total job in candidate field (area), and number of notifications etc)
7. Job application submission: require an interface to upload a resume and type cover letter and submit an application to apply for the specific job.
8. Require an interface to update candidate profile information(personal details, professional details and experience)
9. Candidates should receive a notification of any new posted matching job by employer.
10. Candidates should also receive a notification if the employer has opened their resume, viewed their profile, recruitment selection status or sent any message.
11. Candidates should be able to withdraw their job application.
12. Candidates should be able to apply for multiple jobs according to their interest and skills.
13. Provide an interface to show a list of applied positions. (it should include company name, position, date and time applied, status etc)
14. Candidates should be able to save or bookmark their favorite companies for the future job.
15. Candidates should bookmark their interested location for the job to receive more specific posted job notification.
16. Employee should be able to add or remove their favorite employer (organization) so to receive important alerts for jobs from them.
17. Show list of currently hiring organization (employers).

18. Need account setting window to configure employee account (such as change password, delete account, job location, etc....)
19. Need sign-out button to exit and terminate employee user session data.

2.1.2 Non-functional requirements

Its requirements describe how well it can perform. This basically defines system quality, security, reliability, scalability, and performance. Software requires fulfill below non-functional requirement.

1. Vacancy web applications should take a maximum of 2 seconds to load under normal conditions.
2. Administrator requires saving user activity log (such as login, logout, registered, applied, search performed, posted job etc) with date and time.
3. System should be able to handle at least 15000 concurrent worldwide requests without any performance issues.
4. Sensitive information must be protected using secure authentication and authorization. Properly manage role based user authentication systems and deny unauthorized access to the web. Encrypt sensitive information such as passwords.
5. Require user-friendly interface design and easy navigation system so that the user requires minimum efforts to perform tasks.
6. Application interface should support mobile, tablet, laptop and desktop devices. (It should develop using responsive interface design).
7. Vacancy web applications should be accessed from anywhere in the world and provide fluent user experience.
8. Application must protect common cyber attacks such as SQL Injection, DDoS (Distributed Denial-of-Service) attacks and Cross-site Scripting (XSS)
9. Applications must handle errors (network error, invalid input error, system error etc) and provide proper meaningful error messages to users.
10. System should automatically create a backup at a specific time once in a week.
11. System code must be re-usable, clean and easy to debug. Need useful comments and documentation for essential code.

12. Application requires to maintain user action log (login, posted job, applied for job, date and time etc)
13. Vacancy web applications must support and handle multiple requests at same time.

2.2 Application Process Flow

2.2.1 User Signup/Sign-In flow

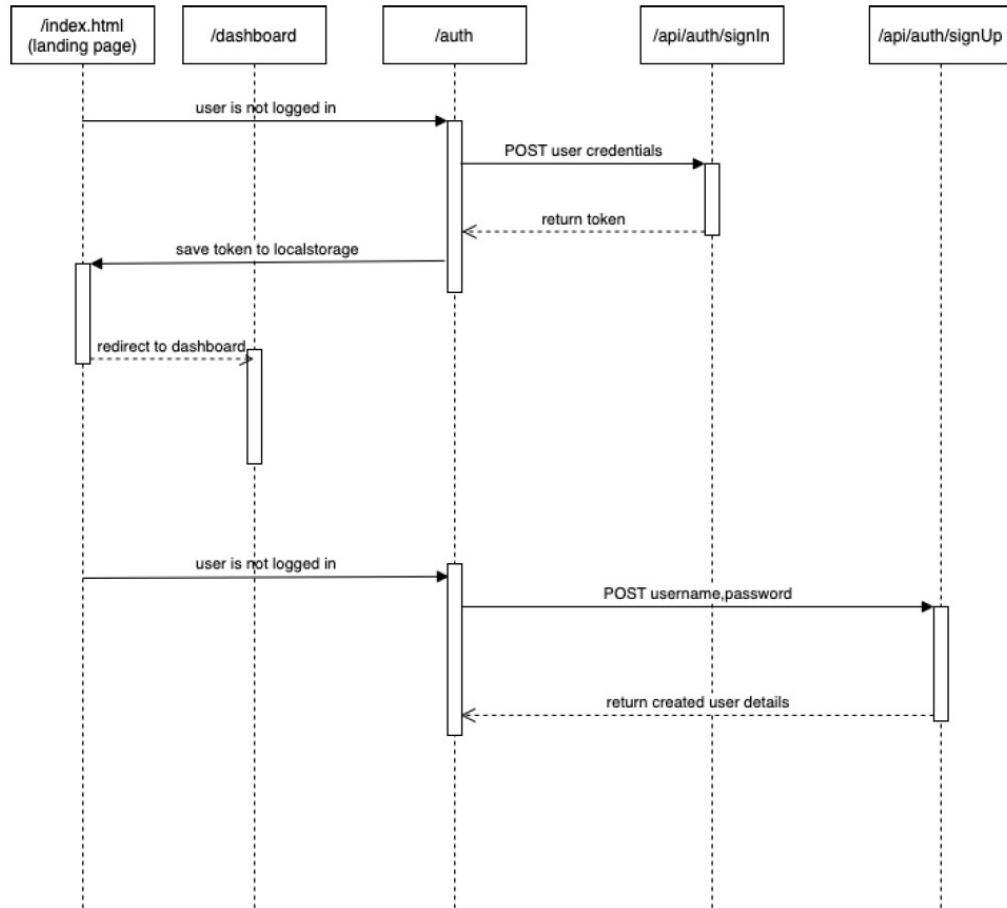


Figure 1: Process flow User Sign-up and Sign -in

2.2.2 Job vacancies process flow

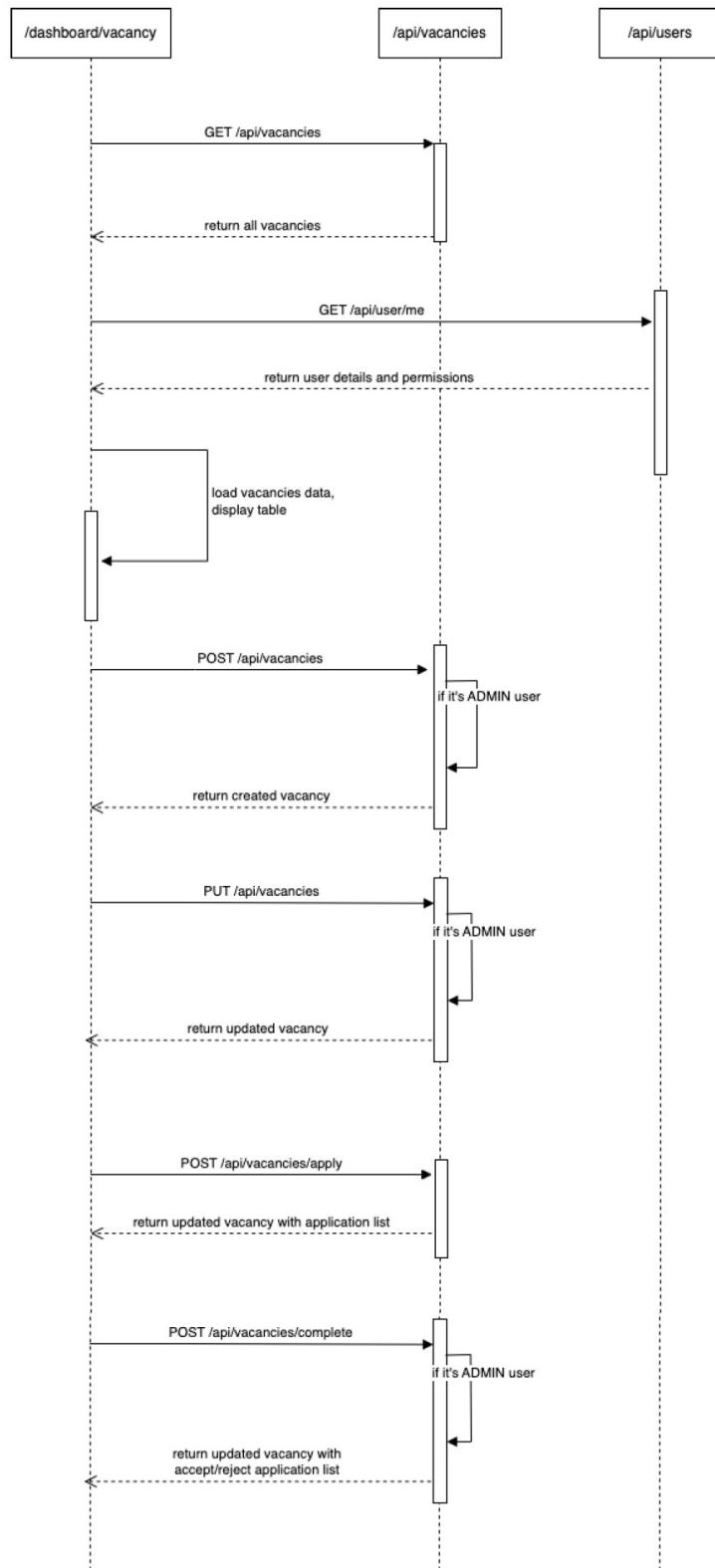


Figure 2: Process flow Job vacancies

2.2.3 User profile management flow

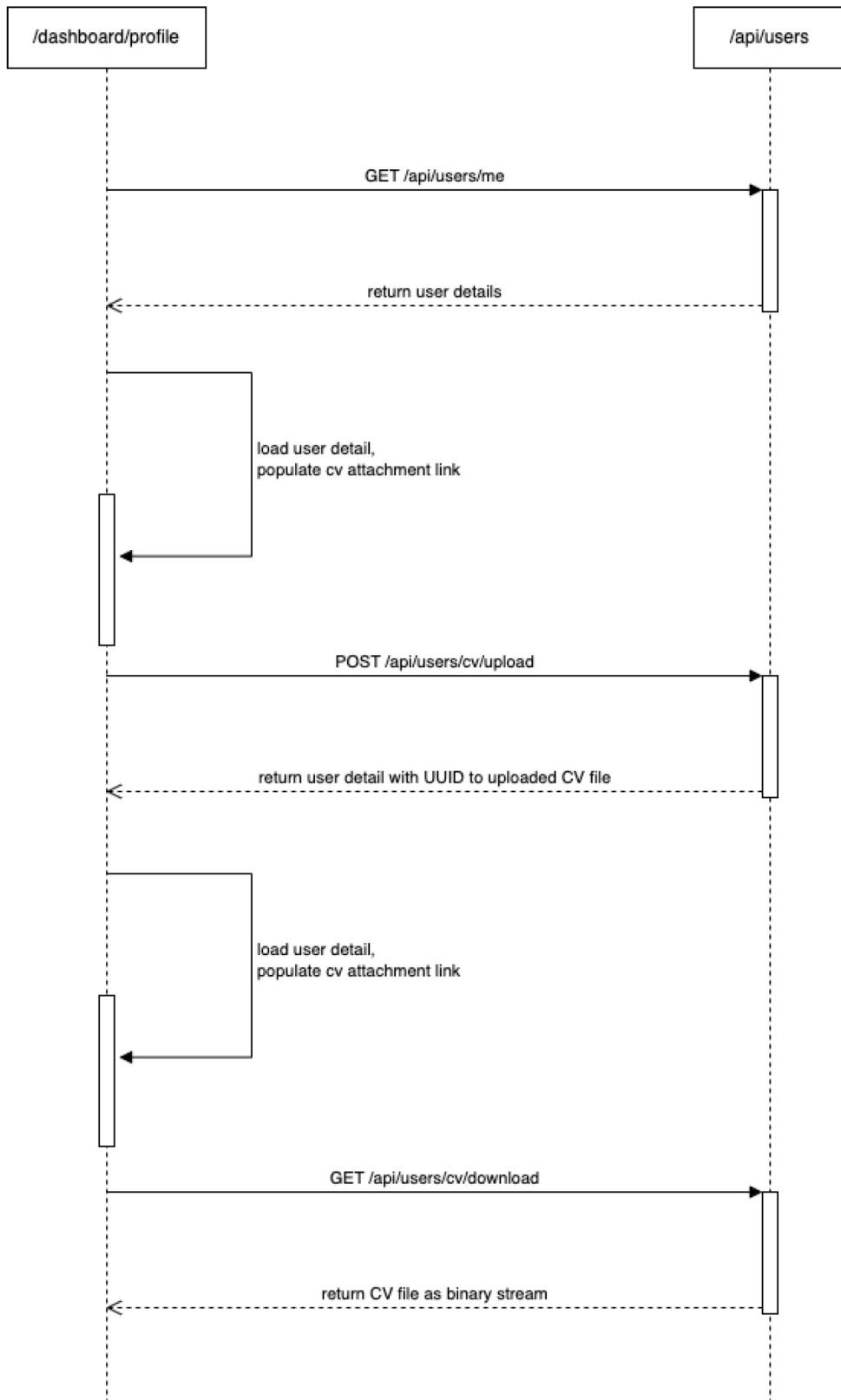


Figure 3: Process flow user profile management

2.3 Use case

The section provides system use case diagram, users (actors), user roles, and use case table.

2.3.1 Use case diagram

The Below use case diagram represents interaction between the user and the system.

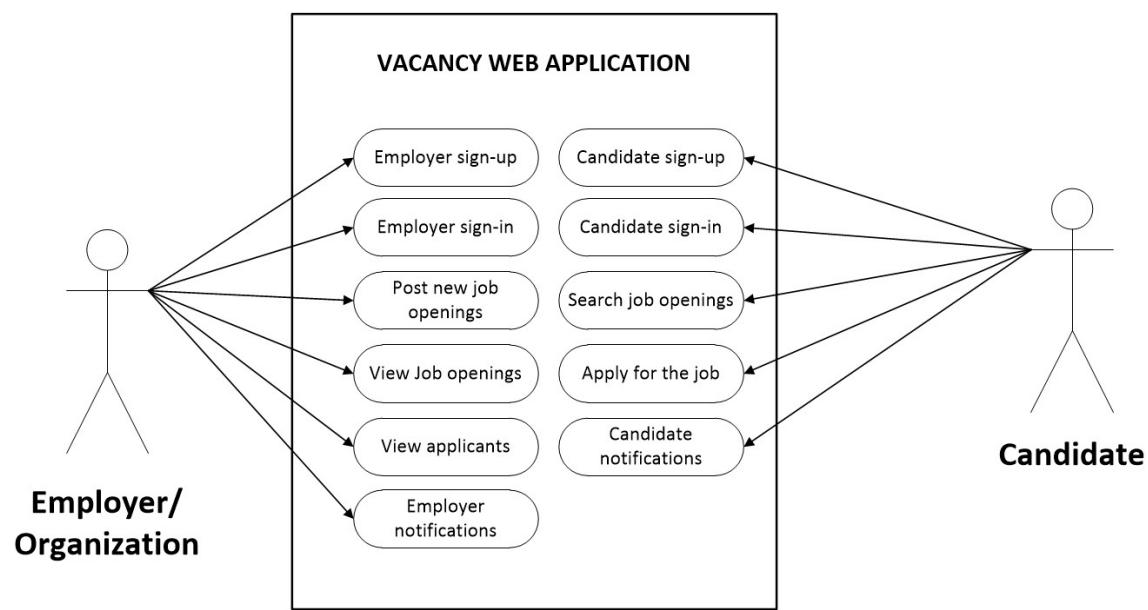


Figure 4: Use-case diagram

2.3.2 Actors

Basically there two actors in our vacancy web application. The bellow table provides user and their roles.

No.	User/Role	Frequency Of Use	Security/Access, Feature used	Additional notes
1	Employer (i.e. any Company or organization)	Occasional	Sign-Up- create a employer account	Only first time registration required
		Occasional	Sign-In- to authenticate	
		Occasional	Post job opening	
		Occasional	Select candidates	
		Occasional	View list of job openings	

		Occasional	Review candidate profile.	
		Occasional	View notifications	
		Occasional	View candidate application	
2	Candidate (i.e. employee)	Frequent	Sign-Up- create a candidate account	Only first time registration required
		Frequent	Sign-In- to authenticate	
		Frequent	Search job openings	
		Frequent	Apply for the jobs	
		Frequent	Review recruiter profile.	
		Frequent	Visit dashboard	
		Frequent	View notifications	

Table 1: System actors

2.3.3 Use case table

The section provides a use case table for vacancy web application, it defines function and interaction for employer and candidates.

UC1	Employer/organization Sign-Up
Actor(s)	Employer
Description	Allow any organization, company or employer to register
Preconditions	Must have the correct company or name, email address and password.
Post conditions	Employer registered successfully.
Main path	1.) Navigate to the employer registration window. 2.) Type organization name 3.) Type organization email address 4.) Type new password 5.) Type confirmation password 6.) Click on submit button 7.) System display message “Registration successful”
Alternate path	6a.) display message names contain special fields. 6b.) display error message if password not match 6c.) display error message if empty fields
Business rules	B1: Check if the employer is already registered.

	B2: Check password creation rules
Non-Functional Requirements	NF1: Store encrypted employer password. NF2: Sign-Up windows should be correctly visible on every device.

Table 2: Use case table 1 Employer/organization Sign-Up

UC2	Employer/organization Sign-In
Actor(s)	Employer
Description	This allows registered employers to authenticate.
Preconditions	1.) Employer must have registered first. 2.) Employer have correct username and password
Post conditions	1.) Employer successfully authenticated and navigated. 2.) Save employer login log.
Main path	1.) Navigate to the employer sign-in window. 2.) Type correct email address or username 3.) Type password 4.) Click on submit button 5.) If the username and password are correct then the employer navigates to the dashboard. 6.) Show error message if invalid username or password
Alternate path	4a.) Display error message if incorrect username and password 4b.) Display error message if empty username or password
Business rules	B1: employer already registered. B2: employer already registered.
Non-Functional Requirements	NF1: Sign-In windows should be correctly visible on every device.

Table 3: Use case table 2 Employer/organization Sign-In

UC3	Employer post new job
Actor(s)	Employer
Description	It allows employers to post new job openings.
Preconditions	1.) Employers must have registered and authenticated to the web. 2.) Employers have correct job details such as title, required experience, required skills, hourly rate, working hours, job type, location and description.

Post conditions	1.) Employers successfully posted new job openings. 2.) Show notification about a newly posted job in the employer account. 3.) Also show notification about new jobs for every candidate.
Main path	1.) Authenticate employer. 2.) Navigate to post new job opening web page 3.) Type correct Job title 4.) Type require skills 5.) Type required experience level 6.) Type hourly rate 7.) Type working hours 8.) Select job type 9.) Select job location 10.) Type job description 11.) Click on the post job button.
Alternate path	11a.) Display error message if incorrect job title, skill, experience, hourly rate, working hours, job type, location or description. 11b.) Display error message if empty fields
Business rules	B1: Check duplicate jobs created or not.
Non-Functional Requirements	NF1: windows should be correctly visible on every device.

Table 4: Use case table 3 Employer post new job opening

UC4	Employer view applicants for job
Actor(s)	Employer
Description	It allows employers to review candidate application requests, resume and select skilled employees for final interview calls.
Preconditions	1.) Employers must have registered and authenticated to the web. 2.) Employers already post new jobs. 3.) Have further communication message (or time slot) for interview calls
Post conditions	1.) Employer downloaded a resume and reviewed client applications. 2.) Show notification that he/she sent notification to employee for interview calls. 3.) Also show notification to the selected candidate that he/she has been selected.

Main path	1.) Employer successfully authenticated using Sign-in page 2.) Navigate to view job opening web page 3.) Review application requests. 4.) Download and review resumes. 5.) Type Interview call message 6.) click on “Recruit” button to send interview call message to the candidate 7.) Show message “ABC has been recruited for the for Data Entry” 8.) Change Row color to green.
Alternate path	6a.) show an error message if the field is empty.
Business rules	B1:
Non-Functional Requirements	NF1: windows should be correctly visible on every device.

Table 5 Use case table 4-Employer view applicants for job

UC5	Candidate search jobs
Actor(s)	Candidate
Description	This allows candidates to search job openings by title, skill, experience, and location. Candidates can apply for the job if he/she finds a relevant job.
Preconditions	1.) Candidates must have registered and authenticated to the web. 2.) Employers already post new jobs. 3.) candidates have search criteria title, skill experience and location details
Post conditions	1.) Candidates have found jobs according to filters.
Main path	1.) Candidate successfully authenticated using Sign-in page 2.) Navigate to view job opening web page 3.) view list of available jobs 4.) Further type filters (i.e. title, skill, experience, and location) 5.) click on search button 6.) display records according to filters 7.) select relevant job and click on “apply for the job” to apply 8.) Candidate navigates to “Apply job” web page.
Alternate path	7a.) show an error message if the field is empty. 7b.) show an error message if a candidate has already applied for a particular job.
Business rules	B1: Candidate should not apply again for same job post

Non-Functional Requirements	NF1: windows should be correctly visible on every device. NF2: Job search should work correctly.
-----------------------------	---

Table 6 Use case table 5-Candidate search jobs

UC6	Candidate apply for job openings
Actor(s)	Candidate
Description	It will allow candidates to write a cover letter and submit their resume for a particular selected job.
Preconditions	1.) Candidates must have registered and authenticated to the web. 2.) Employers already post new jobs. 3.) Candidate had selected a particular job.
Post conditions	1.) Candidate applied and sent a resume for the particular job. 2.) Candidate has covered the latter content. 2.) Candidate has a resume in word or pdf document file.
Main path	1.) Candidate successfully authenticated using Sign-in page 2.) Candidate already selected job and navigated to the “apply job” window. 3.) Type or paste cover letter. 4.) Browse and select latest resume (i.e. word or PDF document file) 5.) click on apply button 6.) display message “You have successfully applied for the given job”
Alternate path	6a.) show an error message if the field is empty. 6b.) show error message if incorrect or invalid data entered in cover letter.
Business rules	B1: Display standard cover letter format in the cover letter text area. B2: Validate document file (i.e. Word or PDF file) correctly uploaded B3: Validate candidate did not select wrong or empty file.
Non-Functional Requirements	NF1: windows should be correctly visible on every device.

Table 7 Use case table 6 Candidate apply for job openings

2.4 Wireframes/Mockup screens

This section provides visual representation of vacancy web applications. It is a basic layout structure into two-dimensional sketches which helps to represent how users will interact with the web app.

Figma project link: <https://www.figma.com/community/file/1478846842989482623>

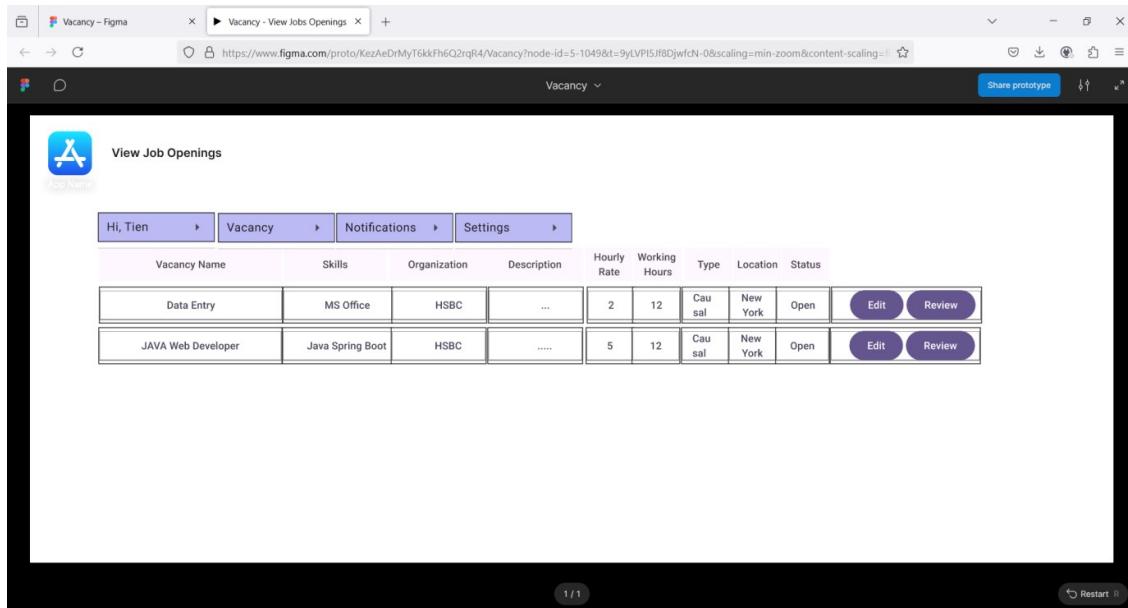
The screenshot shows a 'Sign Up' form on a dark-themed web page. The form includes fields for 'Username' (with placeholder 'Value'), 'Password' (with placeholder 'Value'), and a dropdown menu for 'I am' (with placeholder 'Value'). Below the form are 'Sign-In ? Login Here' and 'Forgot Password?' links, followed by 'Submit' and 'Reset' buttons.

The screenshot shows a 'Review Vacancy' page. At the top, there's a navigation bar with 'Hi, Tien' and links for 'Dashboard', 'Vacancy', 'Notifications', and 'Settings'. The main section is titled 'Vacancy Details' and displays a table with the following data:

Vacancy Name	Skills	Organization	Description	Hourly Rate	Working Hours	Type	Location	Status
Data Entry	MS Office	HSBC	...	2	12	Causal	New York	Open

Below this is a section titled 'Candidate applications' showing three rows of data:

No	Name	Status	CV	Latter	Actions
1	John	PENDING	0235.docx	<button>Accept</button> <button>Reject</button>
2	Rohrson	PENDING	01245.docx	<button>Accept</button> <button>Reject</button>
3	Wilson	PENDING	234235.docx	<button>Accept</button> <button>Reject</button>



2.4.1 Employer sign-up

Using the below window organization or recruiter can easily register themselves by providing name, email and password. After successful sign-up he/she would be able to sign-in.

The wireframe shows a registration form titled "Sign-up (Employer)". It contains four input fields: "Organization name:" with a text input field, "Email address:" with a text input field, "Password:" with a text input field, and "Confirm password:" with a text input field. Below the inputs are two buttons: "Register" and "Reset". At the bottom, there's a link "Sign in? Login here".

Figure 5: wireframe employer Sign-Up

2.4.2 Employer sign-in

This window will be used to authenticate registered organisations by providing email address (username) and password. After successful authentication he/she redirected to the employer dashboard.

The wireframe shows a login form titled "Sign-in (Employer)". It includes fields for "Username/Email" and "Password", and buttons for "Sign-in" and "Reset". A link "Sign up? Register here" is also present.

Sign-in (Employer)	
Username/Email:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Sign-in"/>	<input type="button" value="Reset"/>
<i>Sign up? Register here</i>	

Figure 6: wireframe employer Sign-In

2.4.3 Employer Post new Job openings

This window will be used to post new job openings by providing basic job details such as job title, required skill, experience level, working hours, hourly rate, job type, location and description.

Post new job opening
(employer)

Job title:

Select skills:

Experience level:

Working hours:

Rate hourly:

Job type:

Select location:

Description:

Figure 7: wireframe employer post new job

2.4.4 Employer view Job openings

This window helps to view a list of posted jobs by the employer. Here further they can perform actions like view candidate applications, update job posts or remove.

View Job openings
(Employer)

No	Posted date and time	Job title	Candidates	Skills	Experience	Working hours	Rate hourly	Job type	Job location	Description	Actions
1	15/12/2024 12:01:05	Data Entry	10	MS Office, Computer	1-2 Years	10	2	Casual	NEW YORK	he/she would be able to maintain daily office tasks related to data entry into Excel and Word document files. It will be also useful if the candidate know photo editing.	<input type="button" value="Update"/> <input type="button" value="View applicants"/> <input type="button" value="Remove"/>
2	25/01/2025 11:01:51	Web application	2	Java Spring	1 Years	12	5	Casual	ROTTERDAM	We require web developers who has experience of at least 6 month or 1 year in Java Spring.	<input type="button" value="Update"/> <input type="button" value="View applicants"/> <input type="button" value="Remove"/>
3	25/01/2025 11:01:51	Desktop application developer	1	VB.NET C#, SQL Server	6 Months	12	4	Casual	ROTTERDAM	We require desktop developers who has experience of at least 6 month or 1 year in VB.NET Desktop (C# and VB.NET language).	<input type="button" value="Update"/> <input type="button" value="View applicants"/> <input type="button" value="Remove"/>

Figure 8: wireframe employer view job openings

2.4.5 View Applicants

This window will help to visualize interested candidates for the particular job. An employer can easily review all the applications and resume one by one. He/she will easily analyze and find skilled candidates and further send final interview messages to the selected candidate.

View candidate application
(Employer)

Job title:	Data Entry
Select skills:	MS Office, Computer
Experience level:	1-2 Years
Working hours:	10
Rate hourly:	2
Job type:	Casual
Select location:	NEW YORK

There are a total 3 candidate interested for this job.

No	Applied date and time	Name	Email	Contact No	Experience	Resume	Cover letter	Actions
1	15/12/2024 12:01:05	Smith	smith32656@gmail.com	+31 6 59595811	2 Years	Download	I am writing to express my interest in the [Job Title] position at [Company Name], as advertised on [Where You Found the Job Posting]. With my background in, Read more	<small>Type further communication message and click to recruit</small> Recruit
2	25/01/2025 11:01:51	Williams	williams234@gmail.com	+31 6 32659855	1 Years	Download	In my previous role at [Your Last Job/Experience], I successfully [mention a key responsibility or achievement that relates to the job you're applying for]. Read more	<small>Type further communication message and click to recruit</small> Recruit
3	25/01/2025 11:01:51	Brown	jackson3921@yahoo.in	+31 6 6569548	0 Months	Download	I am confident that my skills, combined with my dedication to excellence and passion for [industry or field], make me a strong candidate for this position. Read more	<small>Booking you in for the selected for the final interview. Please visit our office between 10:00 AM to 5:00PM on Monday 15 Feb 2025.</small> Reject

Figure 9: Employer view candidate requests

2.4.6 Employer notifications

It is windows to view recently performed actions such as posted job, candidate application etc

View Notifications		
(Employer)		
No	Date Time	Description
1	15/12/2024 12:01:05	You have successfully posted a new job for Data Entry.
2	25/01/2025 11:01:51	You have successfully posted a new job for a Web application.
3	25/01/2025 11:01:51	You have successfully posted a new job for a Desktop application developer.
4	25/01/2025 10:03:23	Candidate Abhishek applied for the job Data Entry.
5	01/02/2025 02:23:22	Candidate Rajesh applied for the job Web Application Developer.

Figure 10: wire frame employer notifications

2.4.7 Candidate Sign-Up

The window will help candidates to register themselves by providing name, email and password.

Sign-up	
(Candidate)	
Name:	<input type="text"/>
Email address:	<input type="text"/>
Password:	<input type="password"/>
Confirm password:	<input type="password"/>
<input type="button" value="Register"/> <input type="button" value="Reset"/>	
<i>Sign in? Login here</i>	

Figure 11: wireframe candidate Sign-up

2.4.8 Candidate Sign-In

The window will help to authenticate registered candidates. After successful authentication he/she will navigate to the candidate dashboard.

The wireframe shows a sign-in form titled "Sign In (Candidate)". It includes fields for "Username/email:" and "password:", each with an associated input box. Below the fields are two buttons: "Signin" and "Reset". At the bottom of the form is a link "Sign up? Register here".

Figure 12: wireframe candidate sign-in

2.4.9 Candidate search/view Jobs

The window will help to search for job openings. They can filter jobs by title, skills, experience and location. And further candidates can apply to their suitable jobs.

View Job openings
(Candidate)

Search jobs

Job title:
 skills:
 Experience :
 Location:

Search result....!

No	Posted date and time	Job title	Employer / Organization	Status	Skills	Experience	Working hours	Rate hourly	Job type	Job location	Description	Actions
1	15/12/2024 12:01:05	Data Entry	HSBC	Open	MS Office, Computer	1-2 Years	10	2	Casual	NEW YORK	he/she would be able to maintain daily office tasks related to data entry into Excel and Word document files. It will be also useful if the candidate know photo editing.	<input type="button" value="Apply for the job"/>
2	25/01/2025 11:01:51	Web application	HSBC	Open	Java Spring	1 Years	12	5	Casual	ROTTERDAM	We require web developers who has experience of at least 6 month or 1 year in Java Spring.	<input type="button" value="Apply for the job"/>
3	25/01/2025 11:01:51	Desktop application developer	HSBC	Open	VB.NET C#, SQL Server	6 Months	12	4	Casual	ROTTERDAM	We require desktop developers who has experience of at least 6 month in VB.NET Desktop (C# and VB.NET language).	<input type="button" value="Apply for the job"/>
4	05/01/2025 15:21:15	Artificial Intelligent (AI) Developer	Aviva plc	Open	Machine Learning, Python	1-2 Years	10	2	Casual	PARIS	job description involves designing, developing, and deploying AI solutions to improve business processes. AI jobs often require expertise in machine learning, deep learning, and programming.	<input type="button" value="Apply for the job"/>
5	03/01/2025 02:03:30	Data Science Specialist	Deloitte	Open	Data Collection and analysis	1 Years	12	5	Casual	ISTANBOEL	Need professional who collects large amounts of data using analytical, statistical, and programmable skills	<input type="button" value="Apply for the job"/>
6	02/01/2025 02:03:42	Database Administrator	TCS	Open	MySQL	6 Months	12	4	Casual	BERLIJN	manages an organization's database system, ensuring that it's secure, consistent, and performs well. They also work with developers to improve the database and troubleshoot issues.	<input type="button" value="Apply for the job"/>

Figure 13: wireframe view/search job opening

2.4.10 Candidate apply for the job opening

This window will help to apply for a selected job. Here candidates can further type a cover letter and share the latest resume application.

Apply for the job
(Candidate)

Job title:

Select skills:

Experience level:

Working hours:

Rate hourly:

Job type:

Select location:

Cover letter:

[Your Name]
[Your Address]
[City, State, ZIP Code]
[Your Email Address]
[Your Phone Number]
[Date]

Dear [Hiring Manager's Name],

I am writing to express my interest in the [Job Title] position at [Company Name], as advertised on [Where You Found the Job Posting]. With my background in [Your Field/Industry] and a proven track record of [Key Skills or Achievements], I am excited about the opportunity to contribute to your team and support [Company Name]'s mission.

In my previous role at [Your Last Job/Experience], I successfully [mention a key responsibility or achievement that relates to the job you're applying for]. This experience equipped me with skills in [list relevant skills] and strengthened my ability to [highlight relevant qualities, like problem-solving, teamwork, or leadership].

What excites me most about [Company Name] is [specific detail about the company, like a project, value, or mission that aligns with your goals]. I am eager to bring my expertise in [specific skill or experience] to help [specific goal or challenge the company faces].

I am confident that my skills, combined with my dedication to excellence and passion for [industry or field], make me a strong candidate for this position. I would welcome the opportunity to discuss how my background aligns with your needs in greater detail.

Thank you for considering my application. I look forward to the possibility of contributing to [Company Name]'s success and would be happy to discuss my qualifications further in an interview. I am available at your earliest convenience and can be reached at [your phone number] or via email at [your email address].

Sincerely,
[Your Full Name]

Browse resume:

Figure 14: wireframe candidate apply for job opening

2.4.11 Candidate view notifications

The window will help to view candidate notification such as new job posted, selected by employer etc

View Notifications (Candidate)		
No	Date Time	Description
1	01/02/2024 12:01:05	You have successfully applied for the job Data Entry.
2	02/02/2025 11:01:51	You have successfully applied for job Web application.
3	02/02/2025 03:03:11	TCS company has reviewed your application.
4	07/02/2025 12:02:13	You are selected for a Web Application developer job by TCS. Please check your email/sms for the final interview selection.
5		

Figure 15: wireframe candidate notifications

3 Technical design specification

3.1 Technical stack

3.1.1 Back-end

- Java latest stable version (> v11)
- Maven latest stable version (v3.6)
- Spring-boot (v2.7.4)

3.1.2 Front-end

- Nodejs latest stable version (v18)
- Reactjs (v18.2.0)

3.1.3 Database

- Postgresql lastest stable version

3.2 System UML design

3.2.1 Front-end

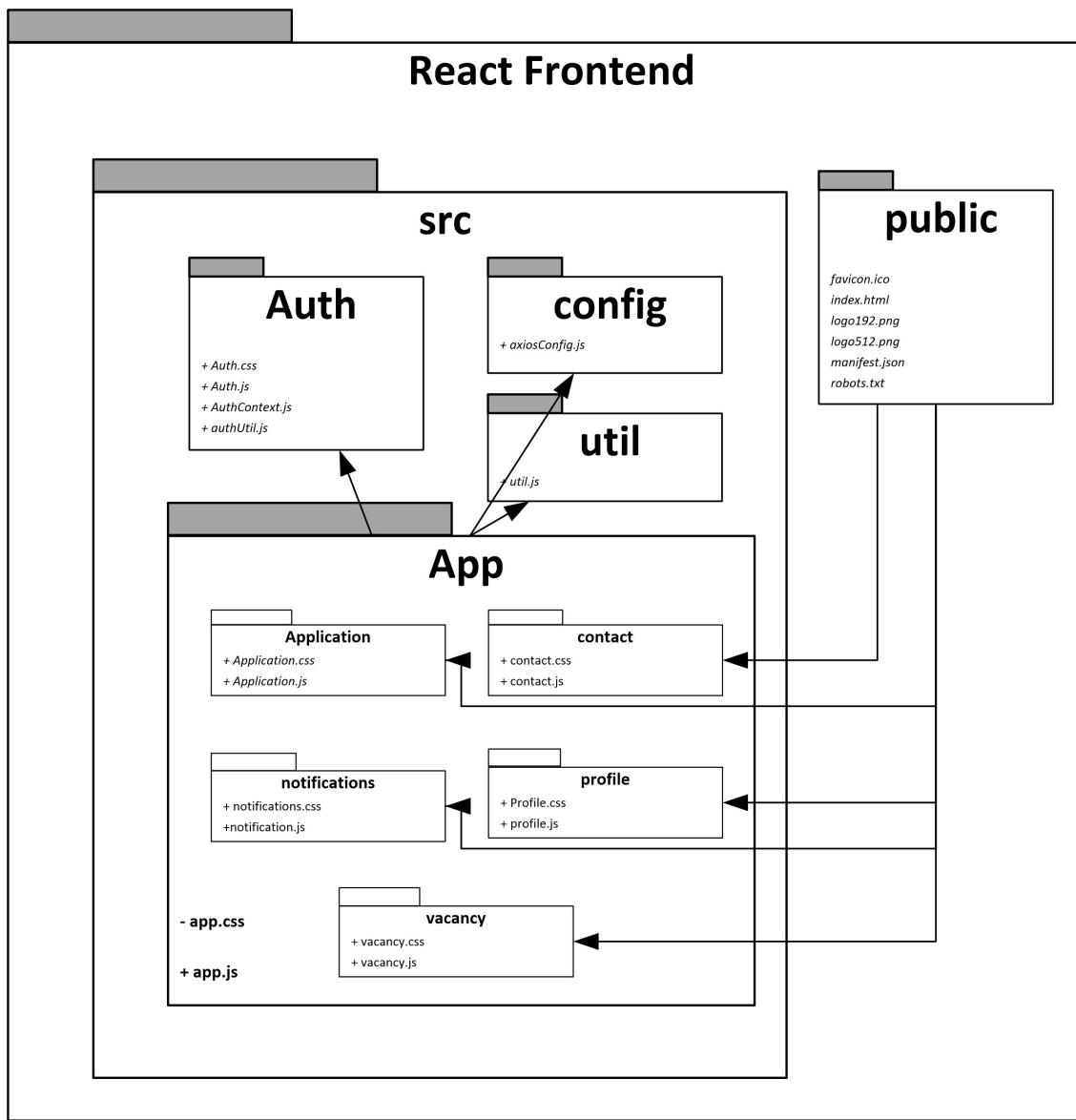


Figure 16: Frontend UML diagram

3.2.2 Back-end

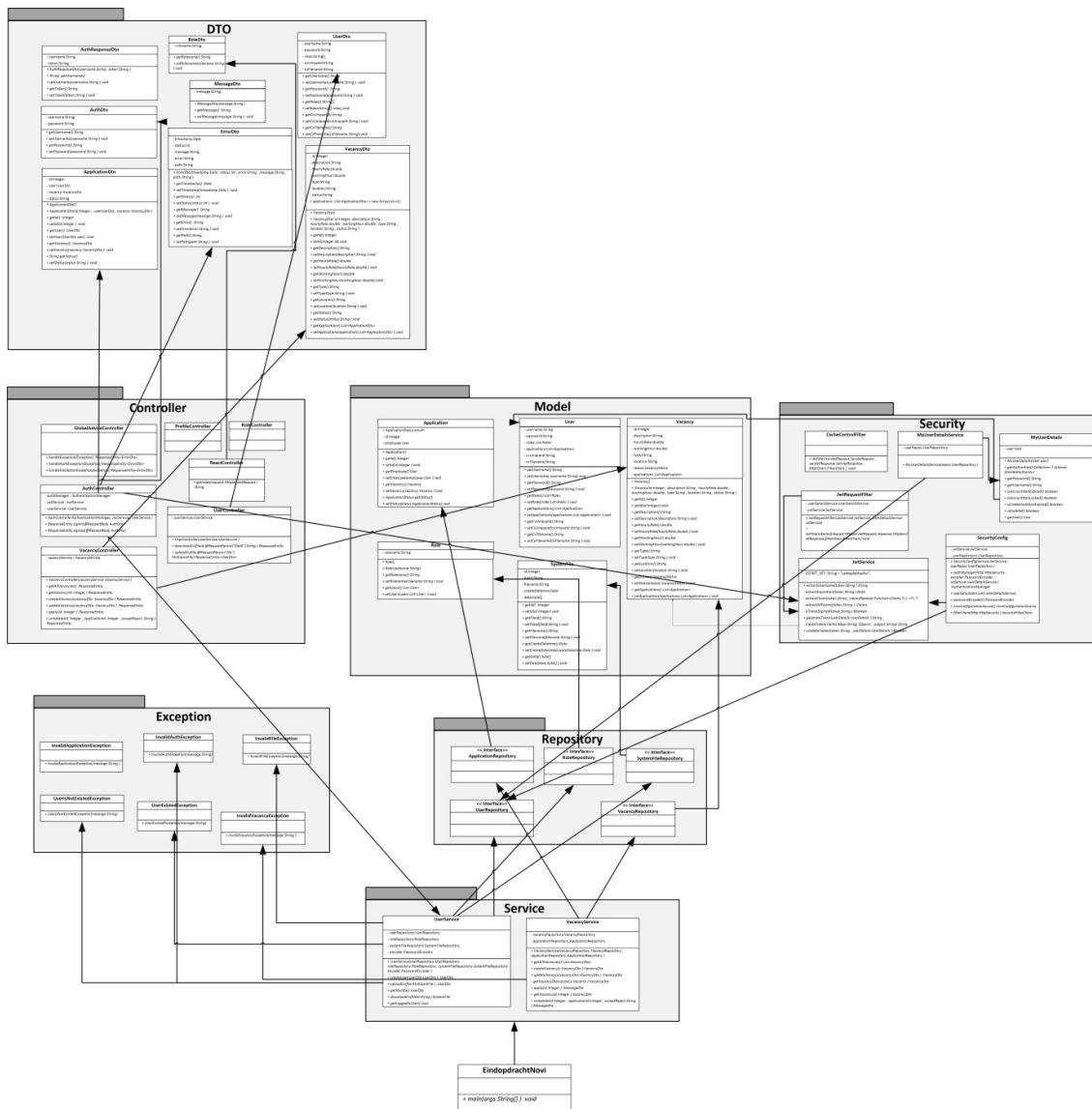


Figure 17: Backend UML Diagram

3.3 API design

3.3.1 Auth api (/api/auth)

3.3.1.1 Sign In API:

- Path: /api/auth/signIn
- Method: POST
- Request: AuthDto

- Response: AuthResponseDto
- Functionality: logged in user and generate jwt token and return in the response.
- Error handling: return UNAUTHORIZED if the user credential is invalid.

3.3.1.2 Sign Up api:

- Path: /api/auth/signUp
- Method: POST
- Request: AuthDto
- Response: UserDto
- Functionality: create user and return detail in the response
- Error handling
 - o return UNAUTHORIZED if the user credential is invalid
 - o Return BAD_REQUEST if the username is already existed

3.3.2 User api (/api/users)

3.3.2.1 Get user info:

- Path: /api/users/me
- Method: GET
- Request: jwt token - header
- Response: Userdto
- Functionality: find user detail by the provided jwt token and return in the response

3.3.2.2 Upload Cv:

- Path: /api/users/upload
- Method: POST
- Request: jwt token – header, cv file – Multipart file
- Functionality: find user detail by the provided jwt token, save the uploaded file into database “systemfile” table as blob data type and return the UUID of uploaded file in the response.
- Error handling:
 - o Return UNAUTHORIZED if the user credential is invalid.

3.3.2.3 Download Cv:

- Path: /api/users/download

- Method: GET
- Request: jwt token – header, UUID of the uploaded file
- Functionality: find the saved file by UUID and return as byte stream to client

3.3.3 Vacancy api (/api/vacancies)

3.3.3.1 *Get all vacancy:*

- Path: /api/vacancies
- Method: GET
- Request: jwt token - header
- Response: VacancyDto
- Functionality: find all vacancies and return in the response

3.3.3.2 *Get all vacancy by id:*

- Path: /api/vacancies/{id}
- Method: GET
- Request: jwt token - header
- Response: VacancyDto
- Functionality: find single vacancy by id and return in the response

3.3.3.3 *Create vacancy:*

- Path: /api/vacancies
- Method: POST
- Request: jwt token - header
- Response: VacancyDto
- Functionality: create vacancy and return in the response
- Error handling:
 - o Return UNAUTHORIZED if the user credential is invalid or it's not ADMIN user.

3.3.3.4 *Update vacancy:*

- Path: /api/vacancies
- Method: PUT
- Request: jwt token - header
- Response: VacancyDto
- Functionality: find vacancy by id and update it and return in the response

- Error handling:
 - o Return UNAUTHORIZED if the user credential is invalid or it's not ADMIN user.

3.3.3.5 - Apply for a vacancy:

- Path: /api/vacancies/apply
- Method: POST
- Request: jwt token – header, vacancy id
- Response: status message
- Functionality:
 - o Check if the vacancy is still in OPEN status, if yes, allow proceeding with the following, otherwise returning 'Already closed' message.
 - o Check if the user is not yet applied for vacancy, if not allow to apply and save the applied record otherwise return 'Already applied' message.

3.3.3.6 Complete decision for the vacancy:

- Path: /api/vacancies/complete
- Method: POST
- Request: jwt token – header, application id, accept/reject status
- Response: status message
- Functionality: find application by id, and set status to ACCEPT or REJECT status
- Error handling:
 - o Return UNAUTHORIZED if the user credential is invalid or it's not ADMIN user.

3.4 Security configuration

- System using spring-boot framework together with spring-boot-security to enabling security around endpoint
- Pseudo-code for security configuration is as per below

The image shows two side-by-side code editors, both titled "SecurityConfig.java".

Top Editor:

```

1 package com.example.eindopdracht.security;
2
3 import ...
4
5 @EnableWebSecurity
6 @EnableGlobalMethodSecurity(prePostEnabled = true)
7 public class SecurityConfig {
8
9     private final JwtService jwtService;
10    private final UserRepository userRepository;
11
12    public SecurityConfig(JwtService service, UserRepository userRepos) {
13        this.jwtService = service;
14        this.userRepository = userRepos;
15    }
16
17    @Bean
18    public AuthenticationManager authManager(HttpSecurity http, PasswordEncoder encoder, UserDetailsService udService) throws Exception {
19        return http.getSharedObject(AuthenticationManagerBuilder.class).AuthenticationManagerBuilder
20            .userDetailsService(udService).DaoAuthenticationConfigurer<AuthenticationManagerBuilder, UserDetailsService>
21            .passwordEncoder(encoder)
22            .and()
23            .build();
24    }
25
26}

```

Bottom Editor:

```

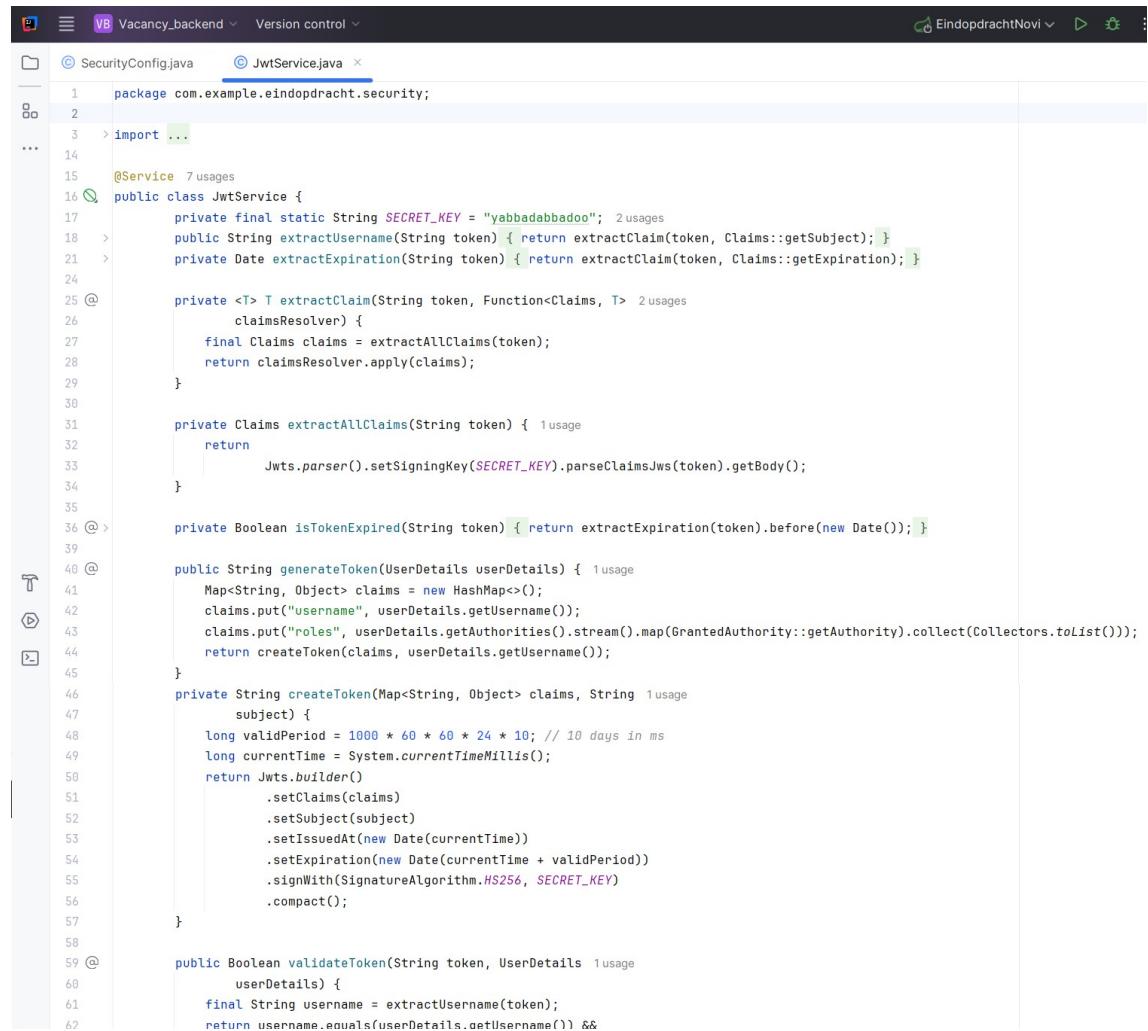
25 public class SecurityConfig {
26
27     @Bean
28     public UserDetailsService userDetailsService() { return new MyUserDetailsService(this.userRepository); }
29
30     @Bean
31     public PasswordEncoder passwordEncoder() { return new BCryptPasswordEncoder(); }
32
33     @Bean
34     public CorsConfigurationSource corsConfigurationSource() {
35         final CorsConfiguration configuration = new CorsConfiguration();
36         configuration.setAllowedOriginPatterns(Arrays.asList("*"));
37         configuration.setAllowedMethods(Arrays.asList("HEAD",
38             "GET", "POST", "PUT", "DELETE", "PATCH"));
39         // setAllowCredentials(true) is important, otherwise:
40         // The value of the 'Access-Control-Allow-Origin' header in the response must not be the wildcard '*' when the request's credentials mode is 'include'.
41         configuration.setAllowCredentials(true);
42         // setAllowedHeaders is important! Without it, OPTIONS preflight request
43         // will fail with 403 Invalid CORS request
44         configuration.setAllowedHeaders(Arrays.asList("Authorization", "Cache-Control", "Content-Type"));
45         final UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
46         source.registerCorsConfiguration(pattern: "/**", configuration);
47         return source;
48     }
49
50     @Bean
51     public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
52         http
53             .cors().and()
54             .httpBasic().disable()
55             .authorizeRequests().ExpressionInterceptUriRegistry
56             .antMatchers(HttpServletRequest.GET, @"/static/**").permitAll()
57             .antMatchers(HttpServletRequest.POST, @"/api/auth/**").permitAll()
58             .antMatchers(@"/api/**").hasAnyAuthority(...authorities: "USER", "ADMIN")
59             .and()
60             .HttpSecurity
61             .addFilterBefore(new JwtRequestFilter(jwtService, userDetailsService()), UsernamePasswordAuthenticationFilter.class)
62             .csrf().disable()
63             .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
64
65         return http.build();
66     }
67
68 }
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87

```

Figure 18: Security configuration setup

- Public resources – under static folder, index.html page is allowed public GET
- Authentication api (/api/auth) is allowed public POST
- CV download api /api/cv/download is allowed public GET
- Other api are secured with jwt token

- Jwt token is secured generated using strong encryption method HS256 by using io.jsonwebtoken jjwt library. Pseudo code for token generation is below



```

1 package com.example.eindopdracht.security;
2
3 > import ...
...
15 @Service 7 usages
16 public class JwtService {
17     private final static String SECRET_KEY = "yabbadabbaoo"; 2 usages
18     >     public String extractUsername(String token) { return extractClaim(token, Claims::getSubject); }
19     >     private Date extractExpiration(String token) { return extractClaim(token, Claims::getExpiration); }
20
21     @ ...
22     private <T> T extractClaim(String token, Function<Claims, T> 2 usages
23         claimsResolver) {
24         final Claims claims = extractAllClaims(token);
25         return claimsResolver.apply(claims);
26     }
27
28     private Claims extractAllClaims(String token) { 1 usage
29         return
30             Jwts.parser().setSigningKey(SECRET_KEY).parseClaimsJws(token).getBody();
31     }
32
33     private Boolean isTokenExpired(String token) { return extractExpiration(token).before(new Date()); }
34
35     @ ...
36     public String generateToken(UserDetails userDetails) { 1 usage
37         Map<String, Object> claims = new HashMap<>();
38         claims.put("username", userDetails.getUsername());
39         claims.put("roles", userDetails.getAuthorities().stream().mapGrantedAuthority::getAuthority).collect(Collectors.toList());
40         return createToken(claims, userDetails.getUsername());
41     }
42
43     private String createToken(Map<String, Object> claims, String subject) {
44         long validPeriod = 1000 * 60 * 60 * 24 * 10; // 10 days in ms
45         long currentTime = System.currentTimeMillis();
46         return Jwts.builder()
47             .setClaims(claims)
48             .setSubject(subject)
49             .setIssuedAt(new Date(currentTime))
50             .setExpiration(new Date(currentTime + validPeriod))
51             .signWith(SignatureAlgorithm.HS256, SECRET_KEY)
52             .compact();
53     }
54
55     public Boolean validateToken(String token, UserDetails userDetails) {
56         final String username = extractUsername(token);
57         return username.equals(userDetails.getUsername()) &&
58
59     }
60
61     @ ...
62 }

```

Figure 19: JWT Token filter

3.5 Sequence diagram

The section provides important three sequence diagrams.

3.5.1 User sign-up/ sign-in

Object and class interaction will be the same in both the user (i.e. employer and candidate) for registration and login in the vacancy web app.

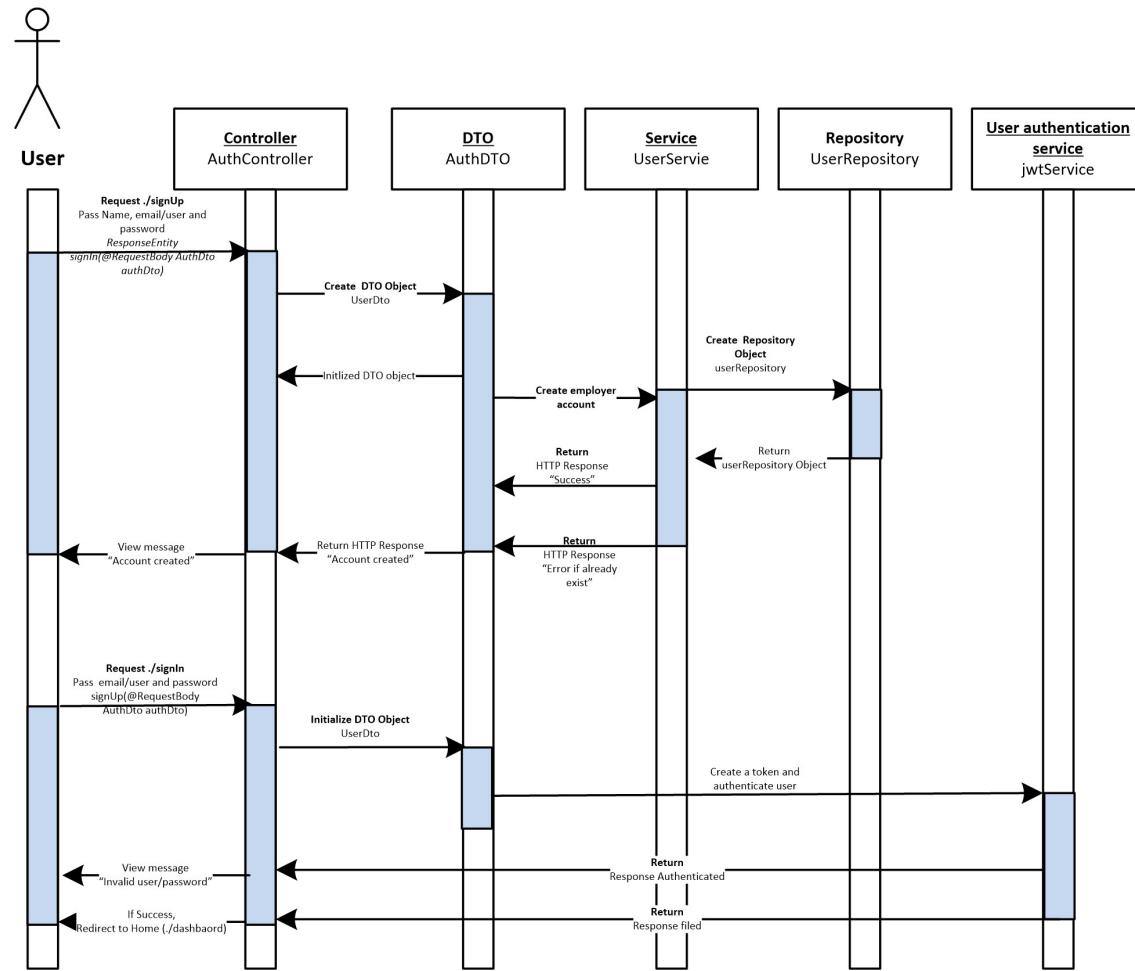


Figure 20: Sequence diagram (User sign-in or sign-up)

3.5.2 Vacancy flow

The diagram represents how objects interact with each other while candidate apply for the job and view vacancy.

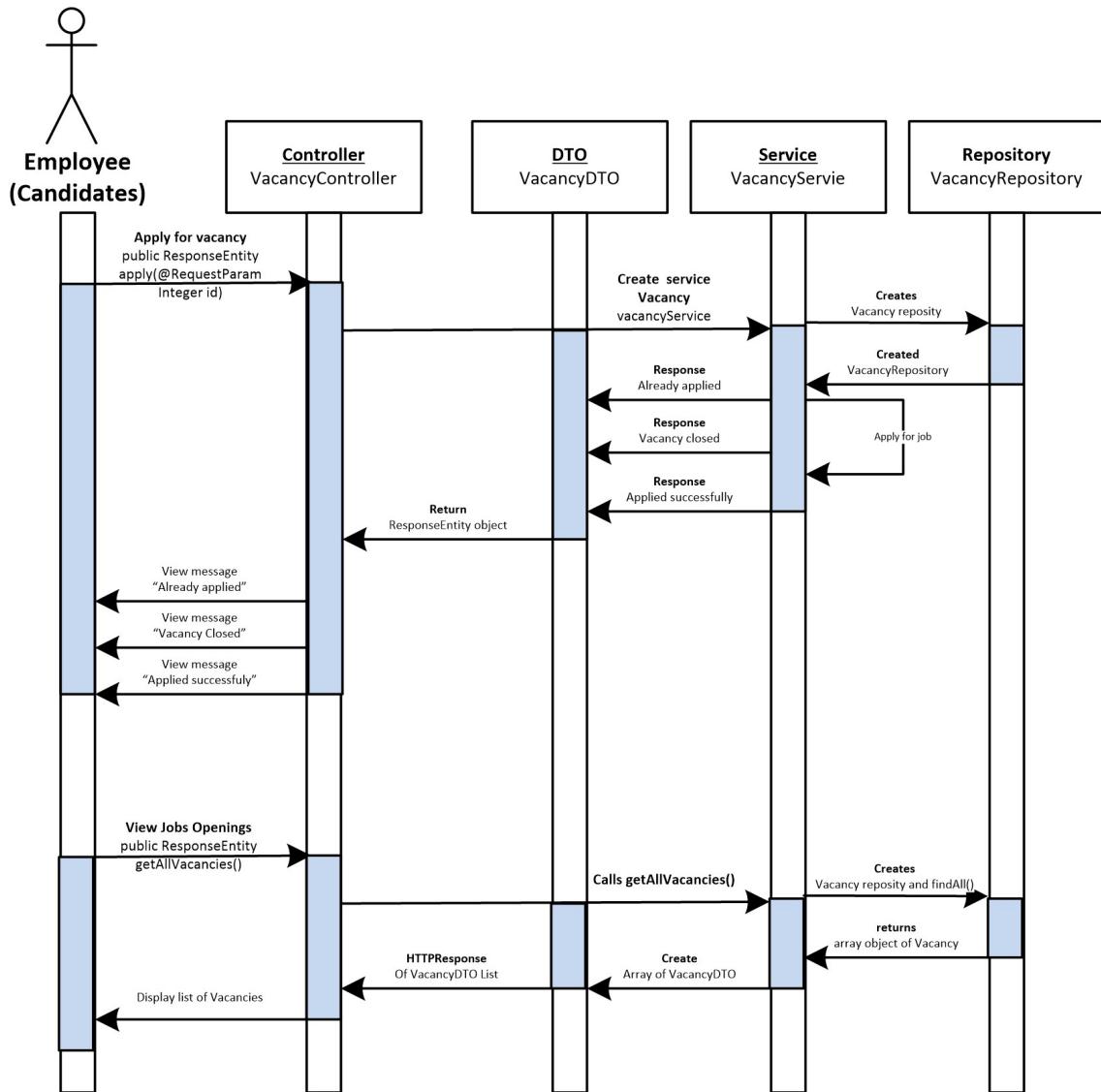


Figure 21: Sequence diagram (View vacancies and apply for job openings)

3.5.3 User profile flow

3.6 Class diagram

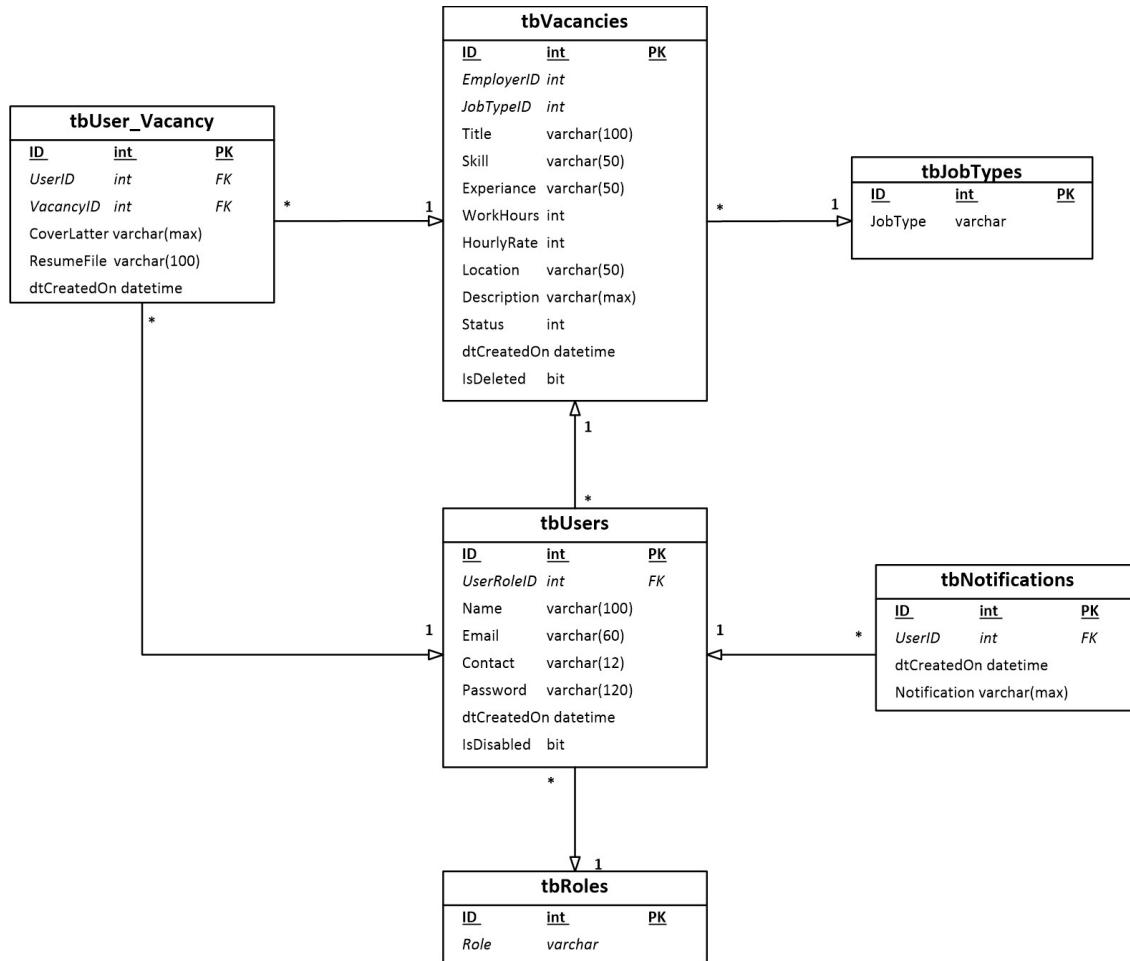


Figure 22: Class diagram

4 Window design

The section gives actual screenshot design for vacancy web application.

4.1 Candidate Sign-Up

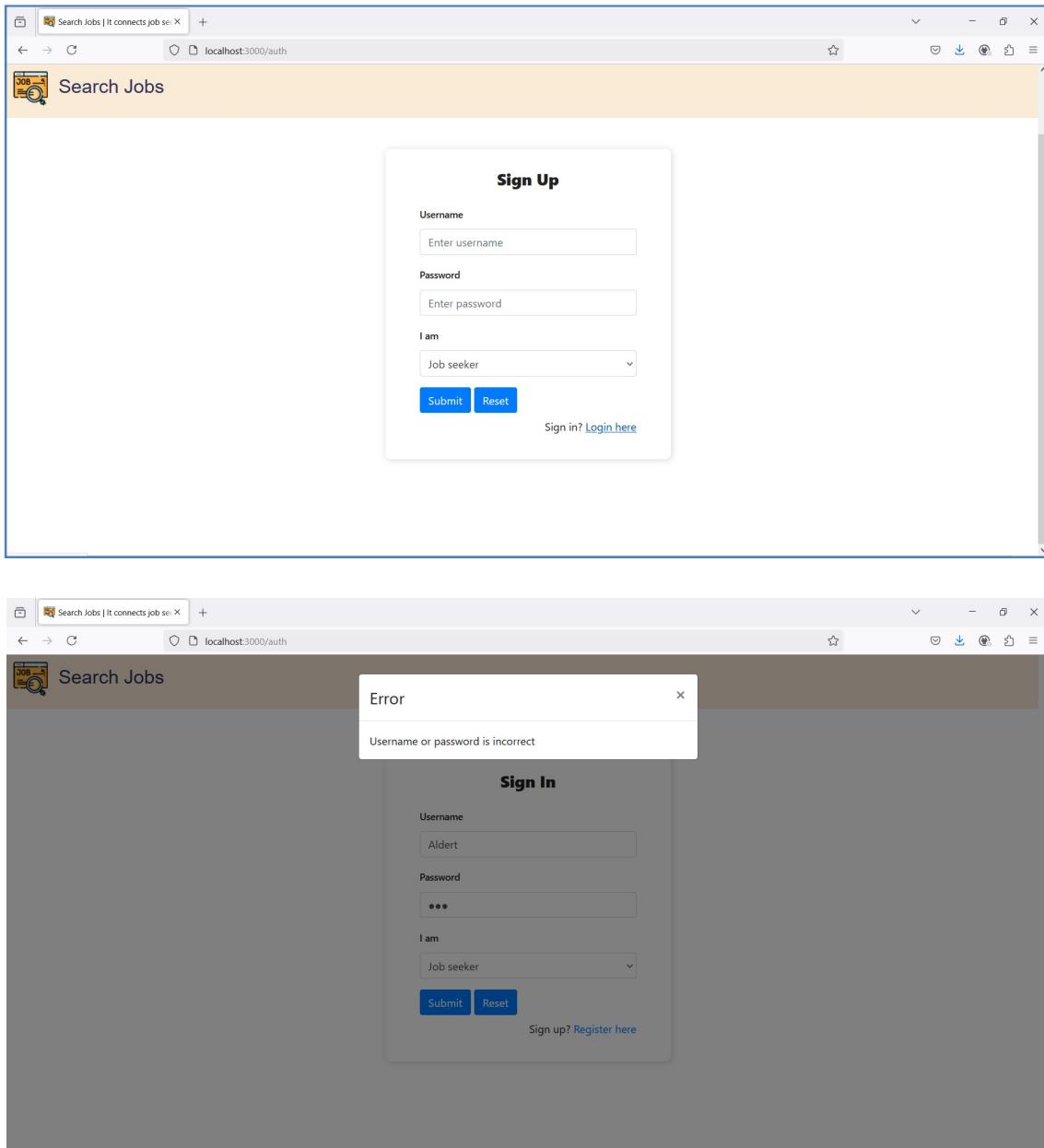


Figure 23: Candidate Sign-up window

4.2 Candidate Sign-In

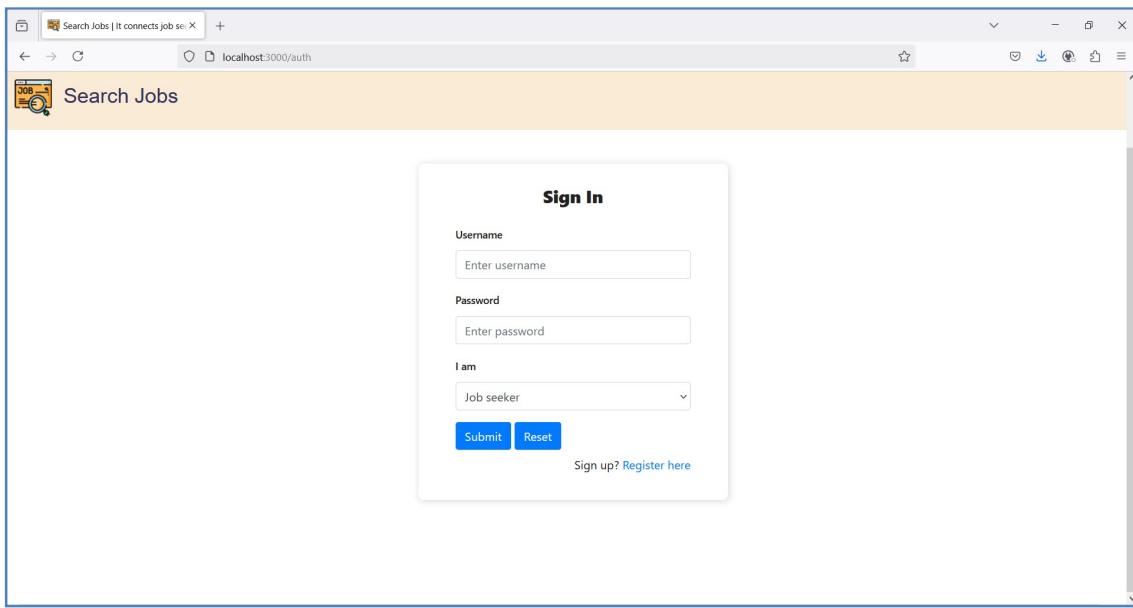


Figure 24: Candidate Sign-in window

4.3 Candidate dashboard

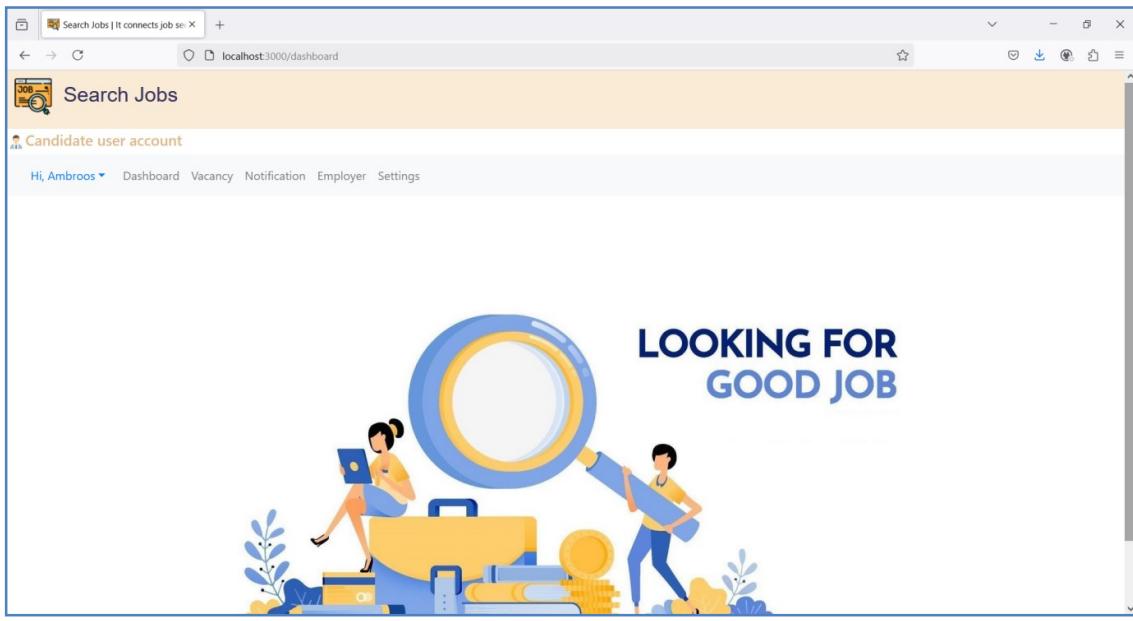


Figure 25: Candidate dashboard

4.4 Candidate Profile

The screenshot shows a web browser window for 'Search Jobs' at localhost:3000/dashboard/profile. The page title is 'Candidate user account'. The navigation bar includes 'Hi, Ambroos' with a dropdown arrow, and links for Dashboard, Vacancy, Notification, Employer, and Settings.

Profile

Username: Ambroos

Your CV (resume): Please upload CV

Upload: Curriculum Vitae (input field) and Browse button

Upload: Upload button

Full name: Ambroos

Email address: Ambroos2sdf@gmail.com

Phone number: +33165689898

Update profile button

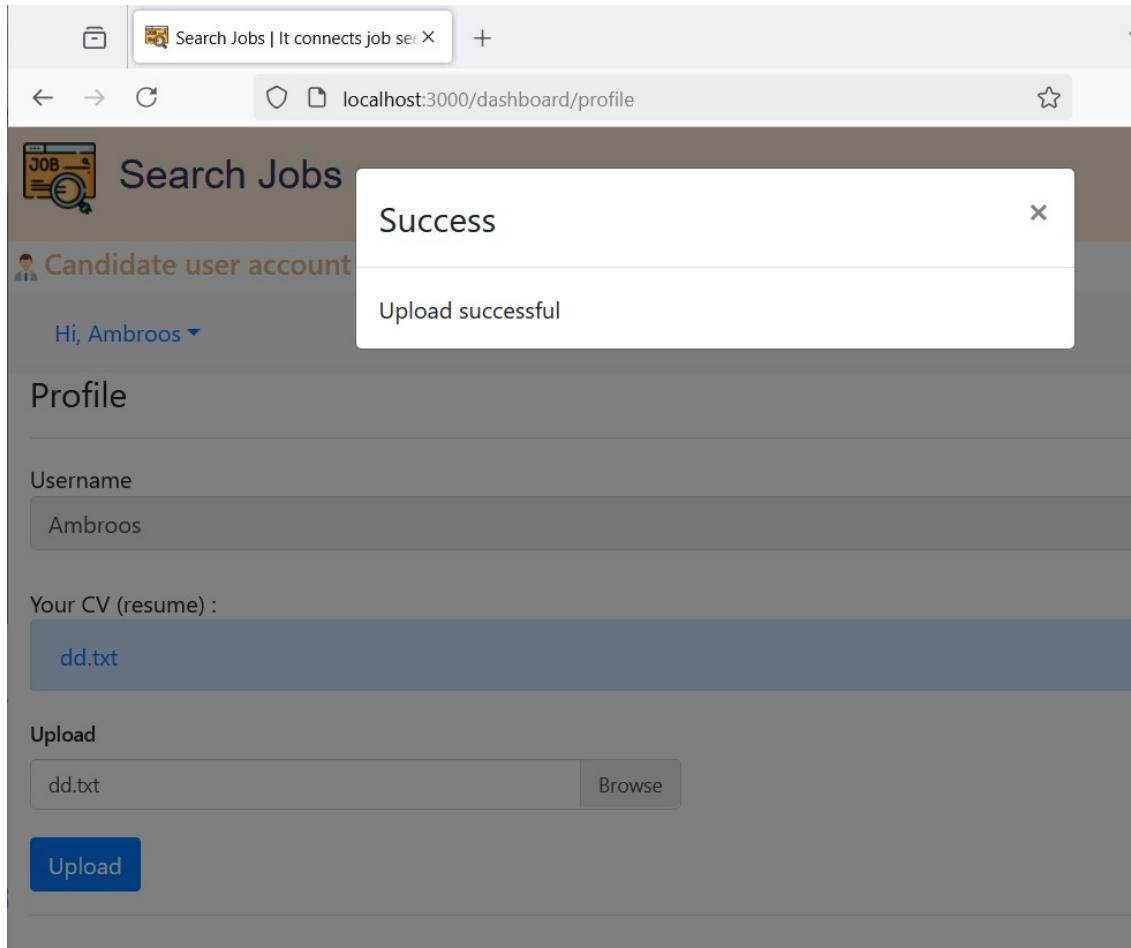
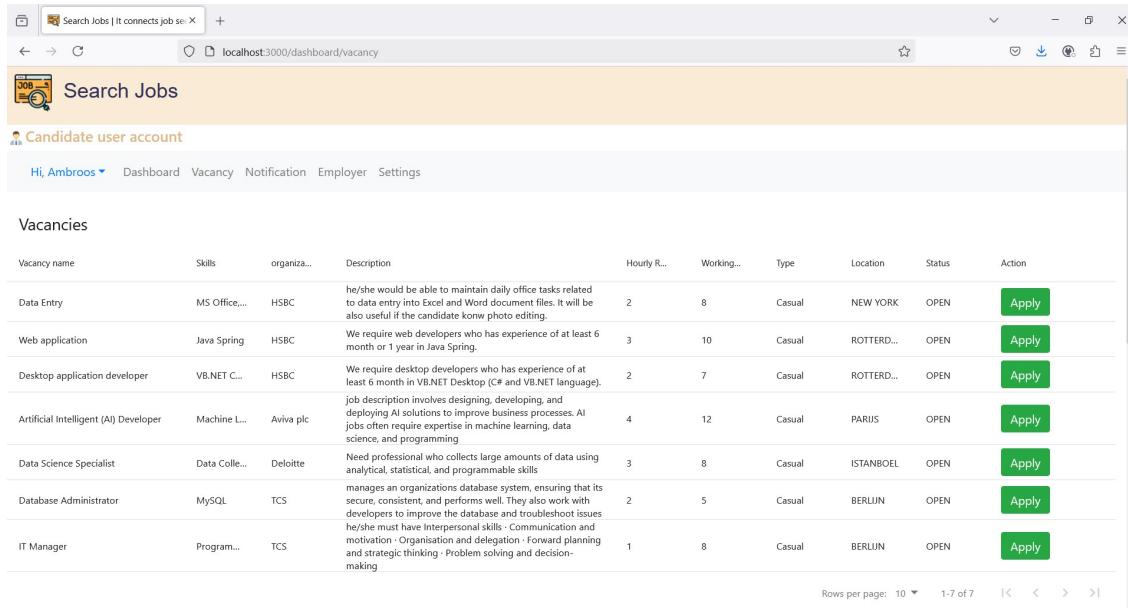


Figure 26: window to upload candidate CV

4.5 Candidate view job openings

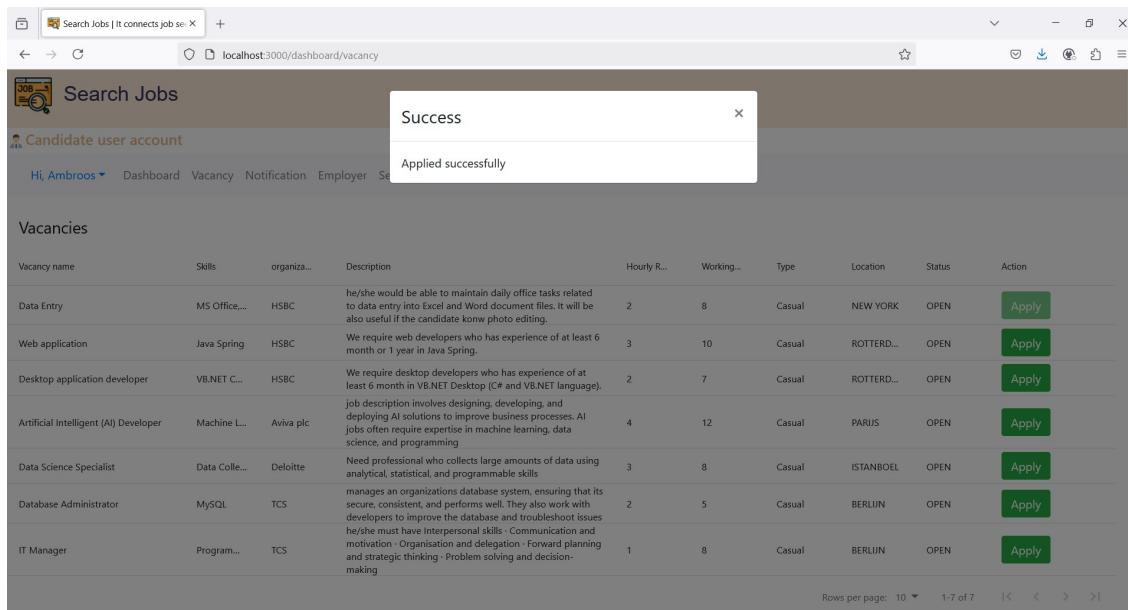


The screenshot shows a web browser window titled "Search Jobs | It connects job seekers and employers". The URL is "localhost:3000/dashboard/vacancy". The page has a header with a user icon, "Candidate user account", and a navigation bar with links: Hi, Ambroos (dropdown), Dashboard, Vacancy, Notification, Employer, Settings.

Vacancies

Vacancy name	Skills	organization	Description	Hourly R...	Working...	Type	Location	Status	Action
Data Entry	MS Office,...	HSBC	he/she would be able to maintain daily office tasks related to data entry into Excel and Word document files. It will be also useful if the candidate know photo editing.	2	8	Casual	NEW YORK	OPEN	<button>Apply</button>
Web application	Java Spring	HSBC	We require web developers who has experience of at least 6 month or 1 year in Java Spring.	3	10	Casual	ROTTERDAM	OPEN	<button>Apply</button>
Desktop application developer	VB.NET C...	HSBC	We require desktop developers who has experience of at least 6 month in VB.NET Desktop (C# and VB.NET language).	2	7	Casual	ROTTERDAM	OPEN	<button>Apply</button>
Artificial Intelligent (AI) Developer	Machine L...	Aviva plc	job description involves designing, developing, and deploying AI solutions to improve business processes. AI jobs often require expertise in machine learning, data science, and programming.	4	12	Casual	PARIS	OPEN	<button>Apply</button>
Data Science Specialist	Data Colle...	Deloitte	Need professional who collects large amounts of data using analytical, statistical, and programmable skills	3	8	Casual	ISTANBOEL	OPEN	<button>Apply</button>
Database Administrator	MySQL	TCS	manages an organizations database system, ensuring that its secure, consistent, and performs well. They also work with developers to improve the database and troubleshoot issues	2	5	Casual	BERLIN	OPEN	<button>Apply</button>
IT Manager	Program...	TCS	he/she must have Interpersonal skills - Communication and motivation - Organisation and delegation - Forward planning and strategic thinking - Problem solving and decision-making	1	8	Casual	BERLIN	OPEN	<button>Apply</button>

Rows per page: 10 ▾ 1-7 of 7 | < < > >|



The screenshot shows the same web application interface as the first one, but with a modal dialog box centered on the screen. The dialog box has a title "Success" and a message "Applied successfully".

The rest of the page content is identical to the first screenshot, showing the list of job vacancies and the "Vacancies" section.

Figure 27: Window to apply job opening

4.6 Candidate Notifications

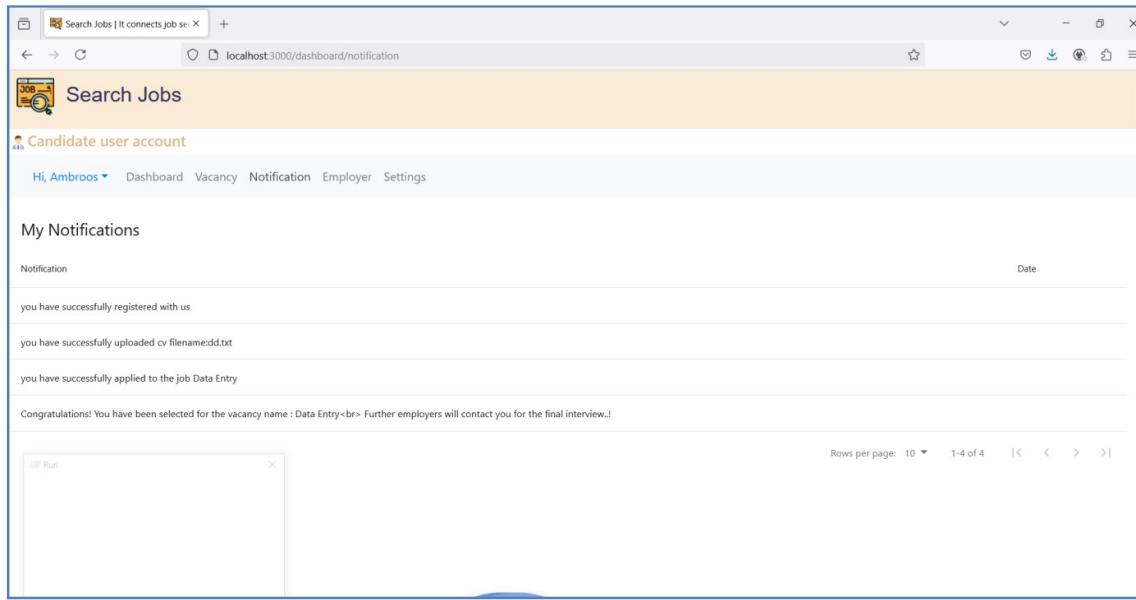


Figure 28: Candidate notifications

4.7 Candidate settings

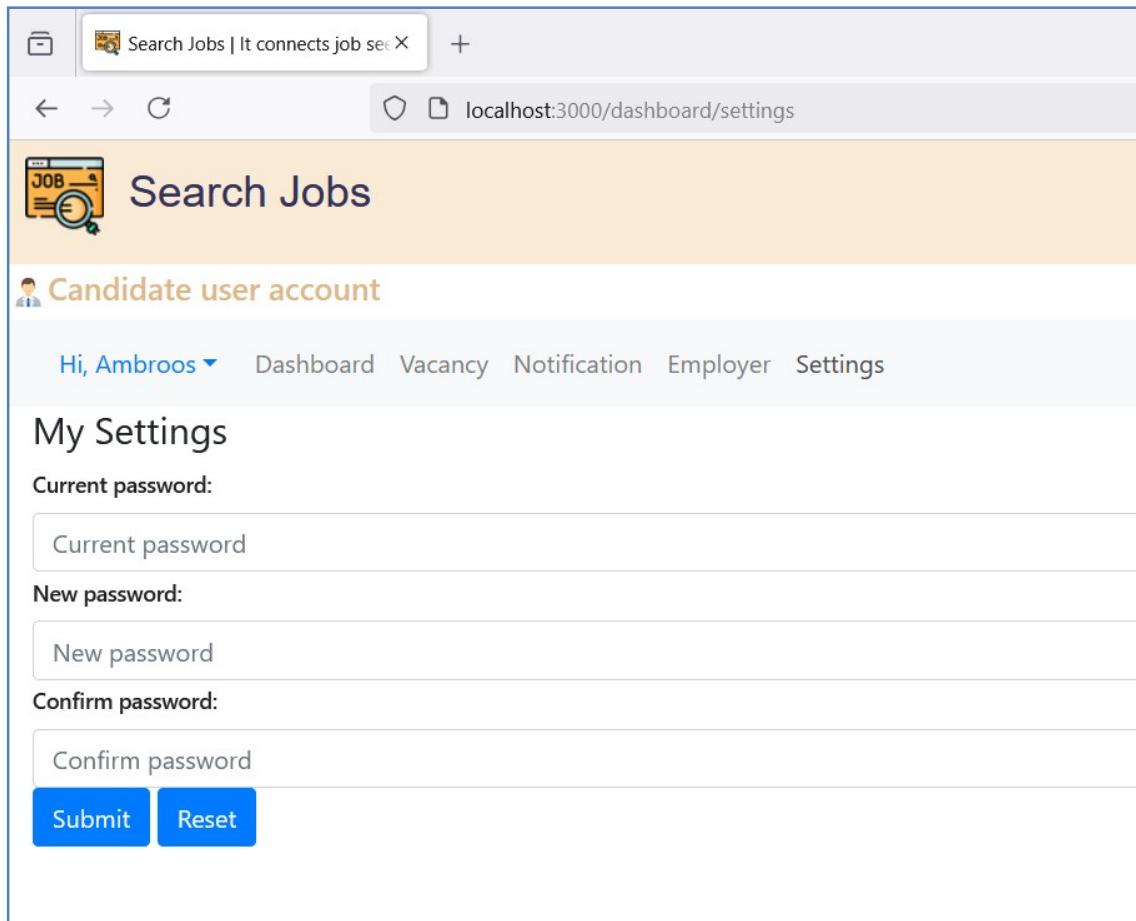


Figure 29: Candidate save settings

4.8 Employer user Signup

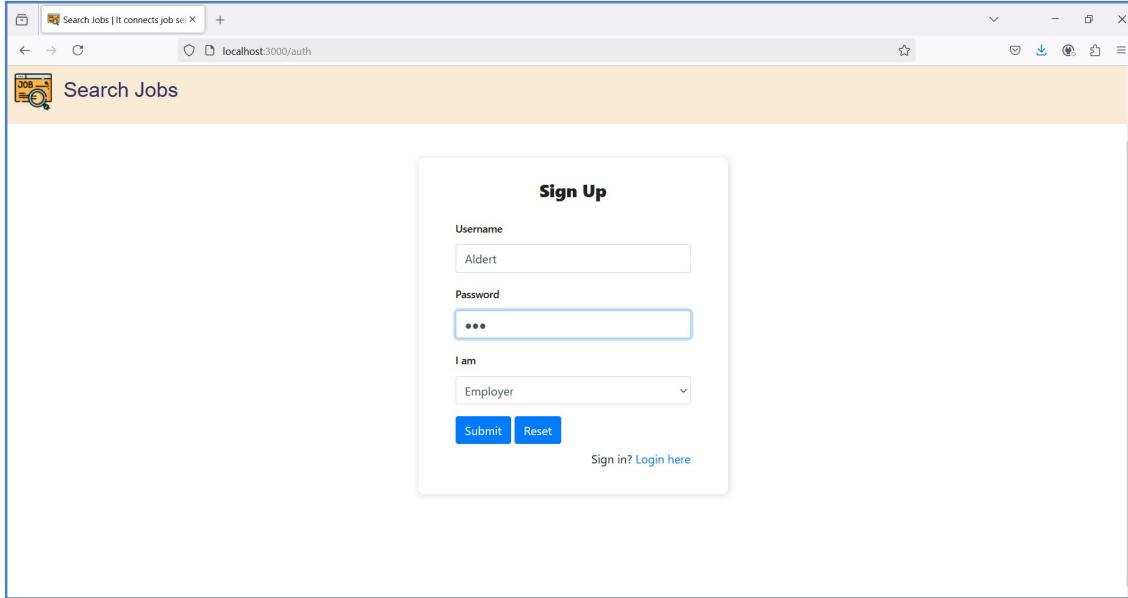


Figure 30: Employer registration window

4.9 Employer user Sign-in

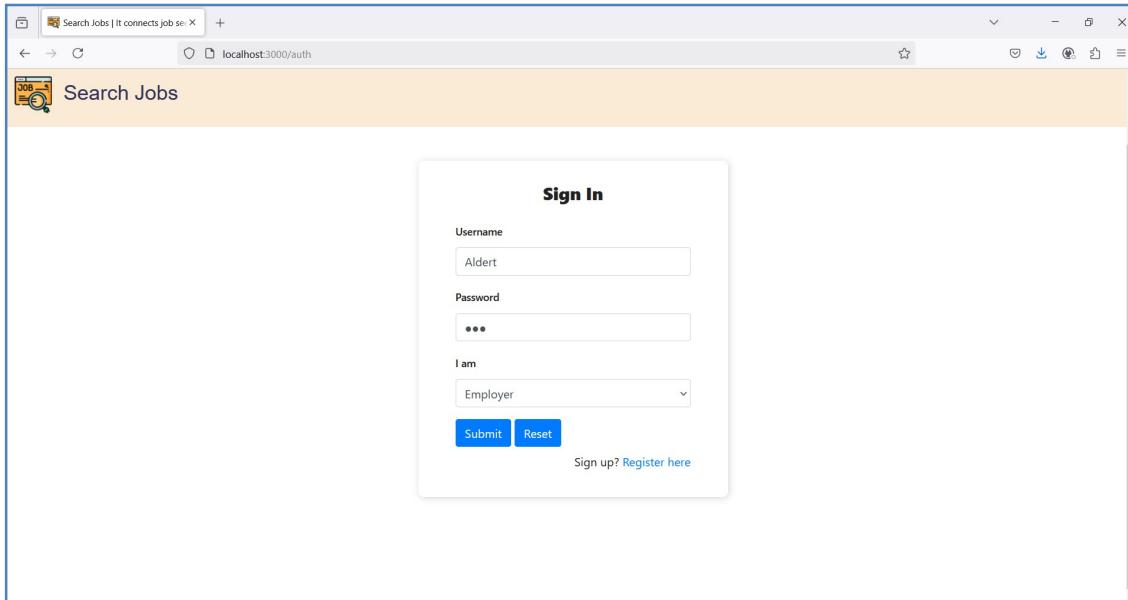


Figure 31: Employer sign-in window

4.10 Employer dashboard

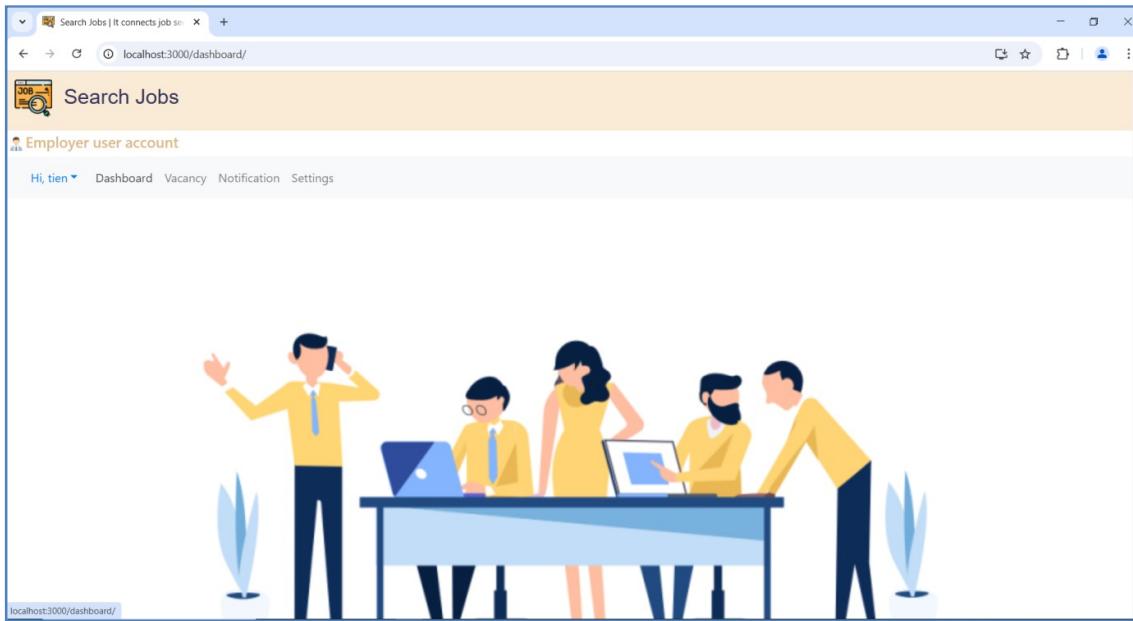
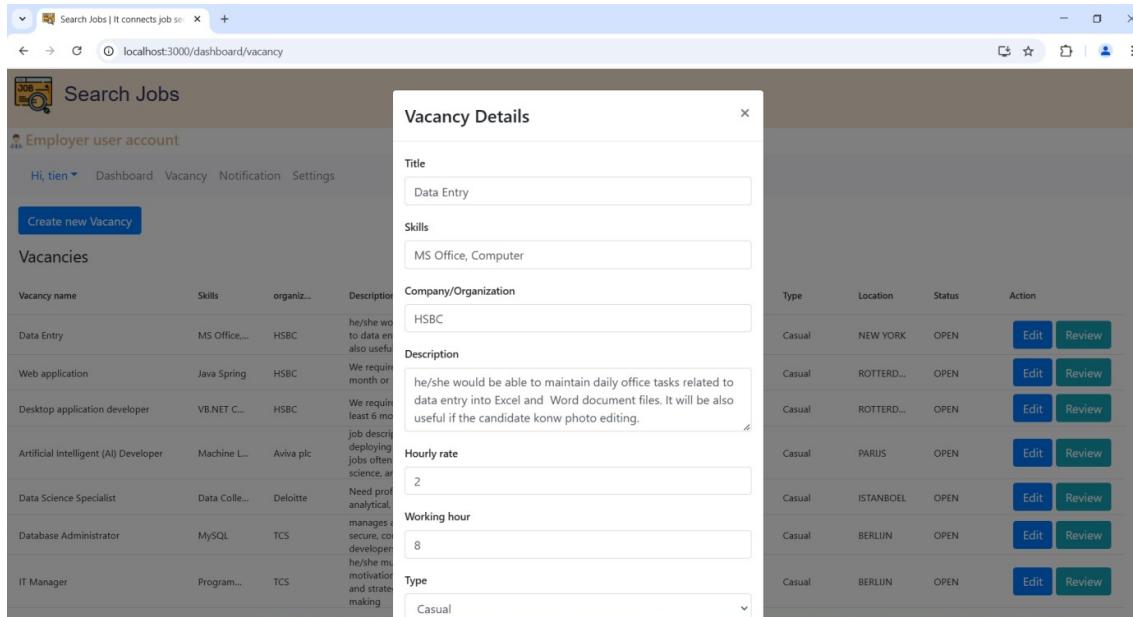


Figure 32: Employer dashboard

4.11 Employer post new vacancy



The screenshot shows a web browser window titled 'Search Jobs | It connects job seekers'. The URL is 'localhost:3000/dashboard/vacancy'. The main page has a sidebar with 'Employer user account' and navigation links: 'Hi, tien' (dropdown), 'Dashboard', 'Vacancy', 'Notification', and 'Settings'. A blue button 'Create new Vacancy' is visible. The main content area is titled 'Vacancies' and lists several job posts with columns: 'Vacancy name', 'Skills', 'organiz...', and 'Description'. A modal window is open on the right side, titled 'HSBC'. It contains fields for 'Description' (text: 'he/she would be able to maintain daily office tasks related to data entry into Excel and Word document files. It will be also useful if the candidate know photo editing.'), 'Hourly rate' (text: '2'), 'Working hour' (text: '8'), 'Type' (dropdown: 'Casual'), 'Location' (dropdown: 'NEW YORK'), and 'Status' (dropdown: 'Open'). At the bottom of the modal are 'Close' and 'Save' buttons.

Type	Location	Status	Action
Casual	NEW YORK	OPEN	<button>Edit</button> <button>Review</button>
Casual	ROTTERDAM	OPEN	<button>Edit</button> <button>Review</button>
Casual	ROTTERDAM	OPEN	<button>Edit</button> <button>Review</button>
Casual	PARIS	OPEN	<button>Edit</button> <button>Review</button>
Casual	ISTANBOEL	OPEN	<button>Edit</button> <button>Review</button>
Casual	BERLIJN	OPEN	<button>Edit</button> <button>Review</button>
Casual	BERLIJN	OPEN	<button>Edit</button> <button>Review</button>

Figure 33: window to post new job

4.12 Employer Job posts

The screenshot shows a web browser window titled 'Search Jobs | It connects job seekers' with the URL 'localhost:3000/dashboard/vacancy'. The interface is identical to Figure 33, but the modal window is closed, and the main 'Vacancies' table is populated with the data from the previous screenshot. The table has columns: 'Vacancy name', 'Skills', 'organiz...', 'Description', 'Hourly ...', 'Workin...', 'Type', 'Location', 'Status', and 'Action' (with 'Edit' and 'Review' buttons).

Vacancy name	Skills	organiz...	Description	Hourly ...	Workin...	Type	Location	Status	Action
Data Entry	MS Office....	HSBC	he/she would be able to maintain daily office tasks related to data entry into Excel and Word document files. It will be also useful if the candidate know photo editing.	2	8	Casual	NEW YORK	OPEN	<button>Edit</button> <button>Review</button>
Web application	Java Spring	HSBC	We require web developers who has experience of at least 6 month or 1 year in Java Spring.	3	10	Casual	ROTTERDAM	OPEN	<button>Edit</button> <button>Review</button>
Desktop application developer	VB.NET C...	HSBC	We require desktop developers who has experience of at least 6 month or 1 year in VB.NET Desktop (C# and VB.NET language). job description involves designing, developing, and deploying AI solutions to improve business processes. AI jobs often require expertise in machine learning, data science, and programming	2	7	Casual	ROTTERDAM	OPEN	<button>Edit</button> <button>Review</button>
Artificial Intelligent (AI) Developer	Machine L...	Aviva plc	Need professional who collects large amounts of data using analytical, statistical, and programmable skills	4	12	Casual	PARIS	OPEN	<button>Edit</button> <button>Review</button>
Data Science Specialist	Data Colle...	Deloitte	manages an organizations database system, ensuring that its secure, consistent, and performs well. They also work with developers to improve the database and troubleshoot issues	3	8	Casual	ISTANBOEL	OPEN	<button>Edit</button> <button>Review</button>
Database Administrator	MySQL	TCS	he/she must have Interpersonal skills - Communication and motivation - Organisation and delegation - Forward planning and strategic thinking - Problem solving and decision-making	2	5	Casual	BERLIJN	OPEN	<button>Edit</button> <button>Review</button>
IT Manager	Program...	TCS		1	8	Casual	BERLIJN	OPEN	<button>Edit</button> <button>Review</button>

Figure 34: employer view jobs

4.13 Employer Review candidate requests

The screenshot shows the 'Employer Review' section of the web application. It displays a table for 'Vacancy Review' with one row:

ID	Description	Hourly Rate	Working Hour	Type	Location	Status
1	he/she would be able to maintain daily office tasks related to data entry into Excel and Word document files. It will be also useful if the candidate konw photo editing.	2	8	Casual	NEW YORK	OPEN

Below this, the 'Applications' section shows two entries:

ID	User	Status	CV	Action
1	bharat	PENDING	dd.txt	<button>Accept</button> <button>Reject</button>
2	Ambroos	PENDING	dd.txt	<button>Accept</button> <button>Reject</button>

The screenshot shows the same 'Employer Review' section as the previous one. The 'Applications' table now includes the 'CV' column for each entry:

ID	User	Status	CV	Action
1	bharat	PENDING	dd.txt	<button>Accept</button> <button>Reject</button>
2	Ambroos	PENDING	No CV resume.txt	<button>Accept</button> <button>Reject</button>

A red circle highlights the 'CV' column for the second application entry, with a red annotation pointing to it: "Employer can download and check candidate CV".

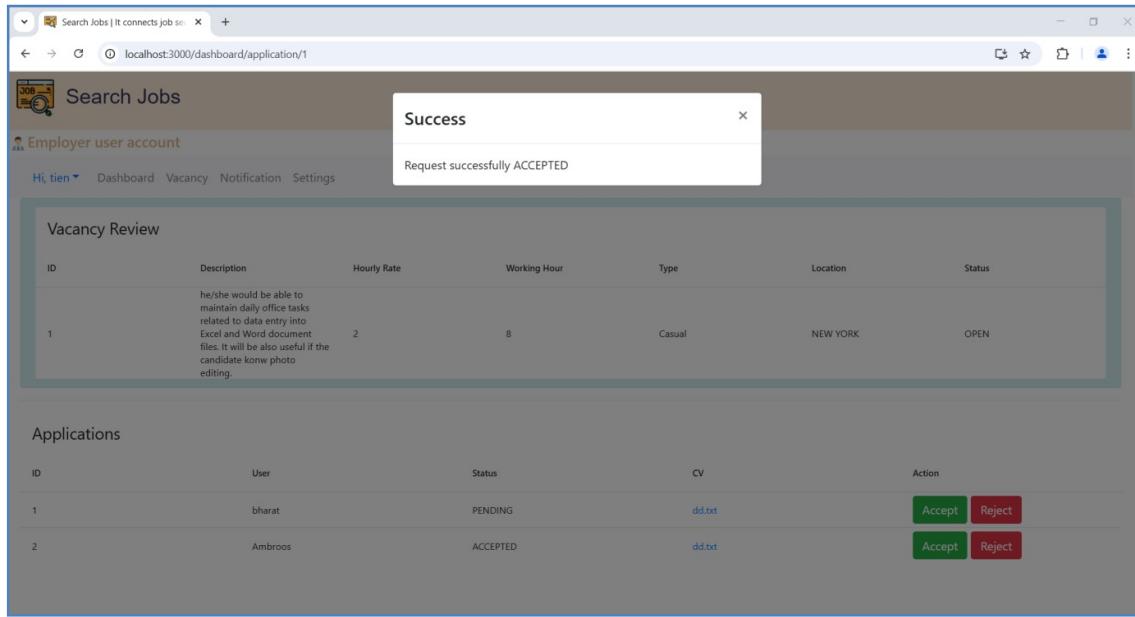


Figure 35: window to review candidate applications

4.14 Employer Notifications

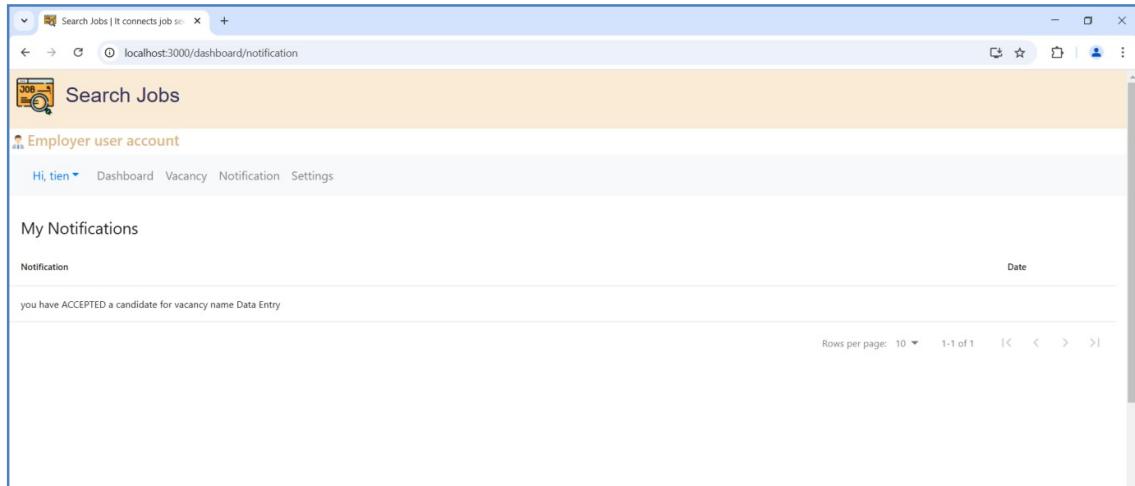


Figure 36: Employer notification window

4.15 Employer settings

The screenshot shows a web browser window with the URL `localhost:3000/dashboard/settings`. The page has a header with a search icon and the text "Search Jobs | It connects job seekers and employers". Below the header, there's a navigation bar with links for "Dashboard", "Vacancy", "Notification", and "Settings". A user profile icon with the name "tien" is visible. The main content area is titled "My Settings". It contains three input fields: "Current password", "New password", and "Confirm password". At the bottom are two buttons: "Submit" and "Reset".

Figure 37: Employer settings

5 Conclusion

The vacancy web application developed using Java Spring Boot and React JS Successfully demonstrates a robust, user friendly, scalable application design to bridge the gap between job employers and job seekers. Through integrating the latest backend technology with responsive front-end (React) technology, this web application offers an efficient platform for managing job posting, users and their interaction.

In conclusion the given project serves as a proof of concept of building full-stack web applications with modern technologies. It can be further extended with additional features to cater industry specific needs, creating valuable tools for employment service and recruitment organizations.

References

dribbble. (2025, 01 01). Job Portal App. Retrieved 01 01, 2025, from dribbble:
https://dribbble.com/tags/job-portal-app?utm_source=chatgpt.com

pinterest. (2025, 01 01). pinterest. Retrieved 01 01, 2025, from Chakri - Job Portal Landing page - v1: <https://in.pinterest.com/pin/706642997789860795/>