Advanced Web Solutions

Report

SID:2022785

MOD004364

Submitted: May 2021

Table of Contents

List Of Figures

Introduction

Web applications, software, and technology have undergone a significant transformation and rapid development due to increased innovation, and their immense benefits to individuals and organizations are vast. The purpose of this project is to develop a web application to generate feedback that will be used by anybody involved in reviewing the application. It hoped that the project would help the company. HappyTech has witnessed an increase in growth and advertises a significant number of positions as writing up letters for every applicant takes a lot of time.

The application is a templet project containing selected features including login page, dashboard, profile editing, generating feedback, sending emails, editing and deleting template, and finally uploading cv/files. The report also describes the wireframe, database design, software design, user stories, and the chosen programming language used to design and build the front and back-end side of the application.

After the application has been fully developed, it would help the company manage the candidate's job application. This will also increase efficiency from the previous method as writing up letters for every applicant takes a lot of time. The application will respond to both the successful and unsuccessful applications by sending them feedback. This report further explains what tools were used to design and develop the application, including xampp as the server, PHP as the scripting language. My SQL was chosen as the database which has been configured and included in the xampp software package.

In conclusion, the application is designed to have a credible user experience and having an easily approachable interface. The implementation and testing of the application are addressed and explained in the last sections of the report.

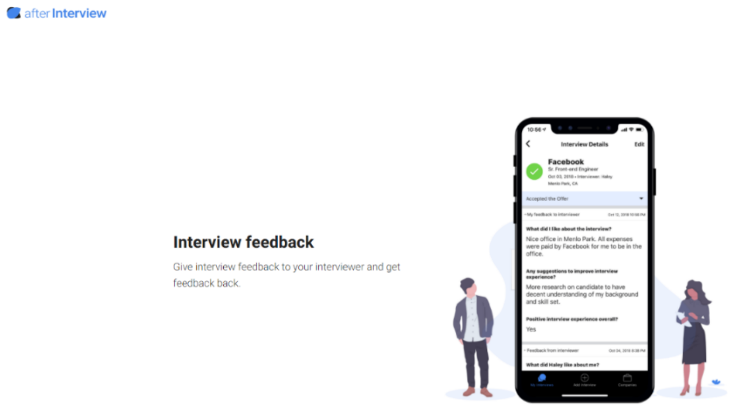
SOFTWARE REQUIREMENTS

1. Google Chrome Browser, v90.0.4430.72 (Google.com, 2017).
2. XAMPP Control Panel, v8.0.1 (Apache Friends, 2019).
3. Atom Code Editor, v1.55.0 x64 (Atom, 2019).

Literature Review

On the internet, there are applications that use the concept of providing candidates with feedback. After Interview and the Recruiter Box are two of the few applications in that category.

After Interview



Interview feedback

Give interview feedback to your interviewer and get feedback back.

Facebook

Sr. Front-end Engineer

Accepted the Offer

What did I like about the interview?

Nice office in Menlo Park. All expenses were paid by Facebook for me to be in the office.

Any suggestions to improve interview experience?

More research on candidate to have decent understanding of my background and skill set.

Positive interview experience overall?

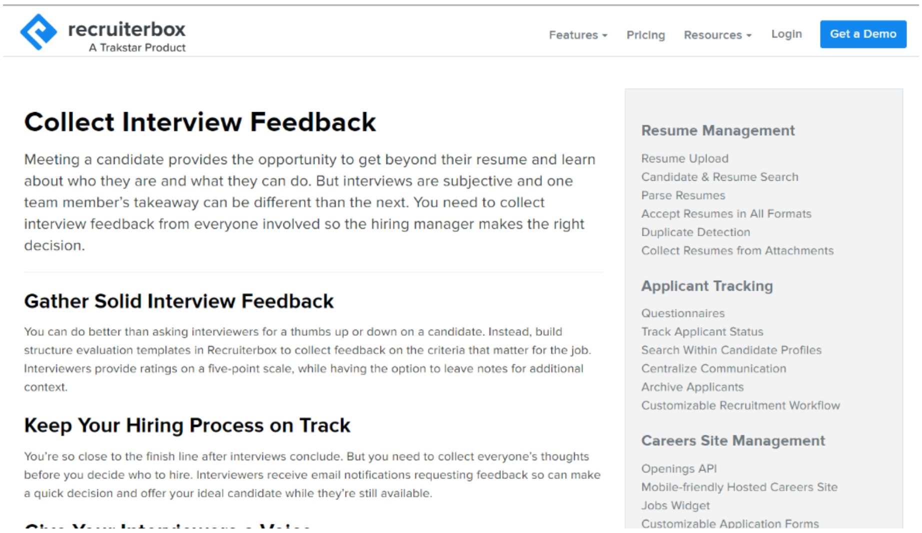
Yes

What did Haley like about me?

Figure 1 After Interview Application

After Interview (2018) is a web application that helps the candidates to improve their skills through the feedback given by the interviewer. This application not only concentrates on helping the candidates to get feedback on their job applications and also concentrates on giving feedback by the candidate to the interviewer. This application also has some features like searching for the interested job, tracking the status of an applied job, and also gives information about the jobs applied by other candidates in past 30 days. One of the useful features is the feedback functionality where it helps the candidate to get feedback on their interview and can get the idea of which companies are giving feedbacks whereas the main drawback is, the candidates who want feedback on their interview need to send the link physically to the interviewer. This could be an issue for candidates when they face any technical issues. To overcome this issue, once the candidate adds the external URL of the job post to the form, it's better to pull the interviewer mail from the post which collects the interviewer's details and sends mail automatically.

Recruiter Box



Collect Interview Feedback

Meeting a candidate provides the opportunity to get beyond their resume and learn about who they are and what they can do. But interviews are subjective and one team member's takeaway can be different than the next. You need to collect interview feedback from everyone involved so the hiring manager makes the right decision.

Gather Solid Interview Feedback

You can do better than asking interviewers for a thumbs up or down on a candidate. Instead, build structure evaluation templates in Recruiterbox to collect feedback on the criteria that matter for the job. Interviewers provide ratings on a five-point scale, while having the option to leave notes for additional context.

Keep Your Hiring Process on Track

You're so close to the finish line after interviews conclude. But you need to collect everyone's thoughts before you decide who to hire. Interviewers receive email notifications requesting feedback so can make a quick decision and offer your ideal candidate while they're still available.

Resume Management
Resume Upload
Candidate & Resume Search
Parse Resumes
Accept Resumes in All Formats
Duplicate Detection
Collect Resumes from Attachments

Applicant Tracking
Questionnaires
Track Applicant Status
Search Within Candidate Profiles
Centralize Communication
Archive Applicants
Customizable Recruitment Workflow

Careers Site Management
Openings API
Mobile-friendly Hosted Careers Site
Jobs Widget
Customizable Application Forms

Figure 2 Recruiter Box Application

Recruiter Box (2011) application is a bit similar web application that uses feedback functionality. But the main aim of the application is to get feedback from persons involved in this process. But those feedbacks are used by the Human Resource team to get the idea of each candidate's strengths and weakness and get the rating on the candidate performance. This helps the company to select the right candidate for their vacancies. In this application, the notification feature helps HR to select the right candidate within a short time without facing any delays. This application can be improved by sending feedback to all people involved including the candidates. For example, sending the same feedback to the candidates will help them to improve their skills before attending another interview in the future.

Framework

CodeIgniter (2019) is the Powerful PHP Server-Side Framework to build dynamic web applications. This Framework is easy to set up and uses a small footprint with built-in libraries which helps the developer to build the applications faster and improves application performance by reducing the server time and bandwidth usage. This framework encourages developers to make use of the MVC and HMVC concepts in their application but it’s not mandatory. And the configuration of the CodeIgniter is much easier when compared with other frameworks. And also, this framework provides an inhibitor for error-handling along with prevention filters that tackle security issues like CRSF and XSS attacks which are considered as the most common and effective attacks used by hackers, and these attacks can compromise the whole system.

Software Design

To implement the user stories into the real-world web application, the design and the development parts are divided into categories. And this chapter will give an insight into how the wireframe, software, and database are designed.

Wireframe

This section gives more details about the blueprint of the web application.

Homepage/Landing Page

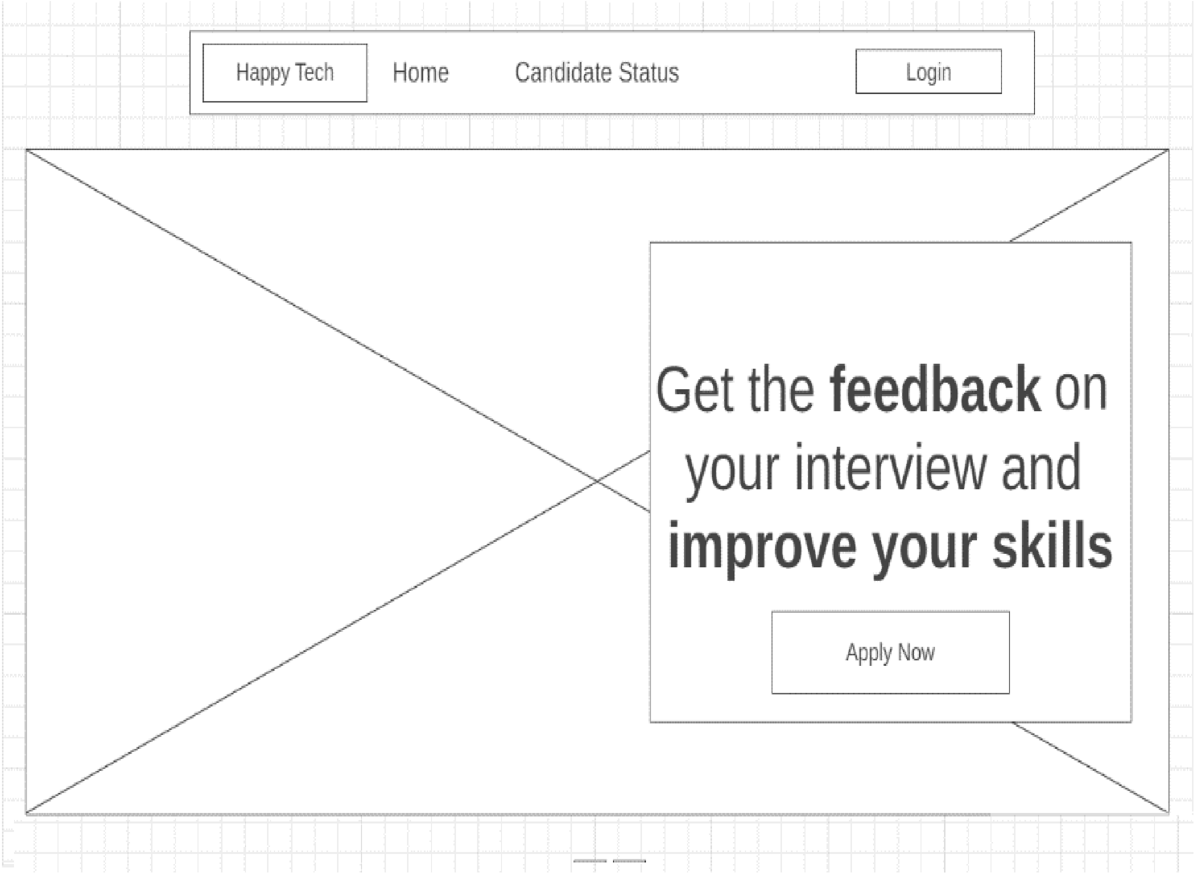


Figure 3 HomePage Wireframe

The homepage consists of features like Home, Candidate Status, Apply Now, login and Register. Every feature on the homepage is public and can be accessed by all users. To use the main features of an application, users should log in or register whereas the candidates can access features like applying for a job and check the status online without the need of registration.

Login and Registration Page

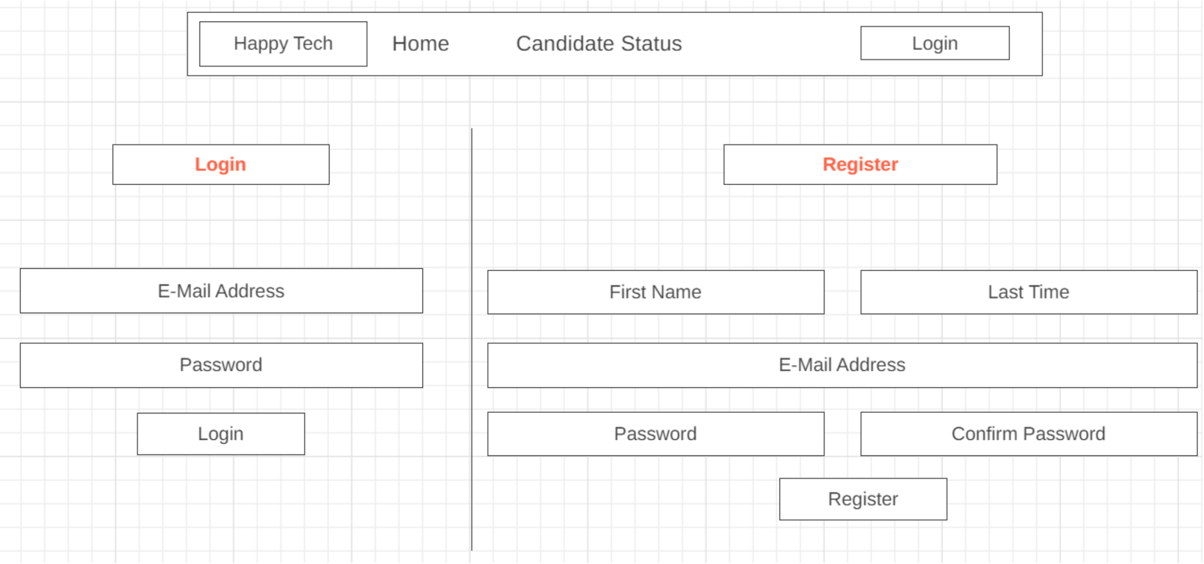
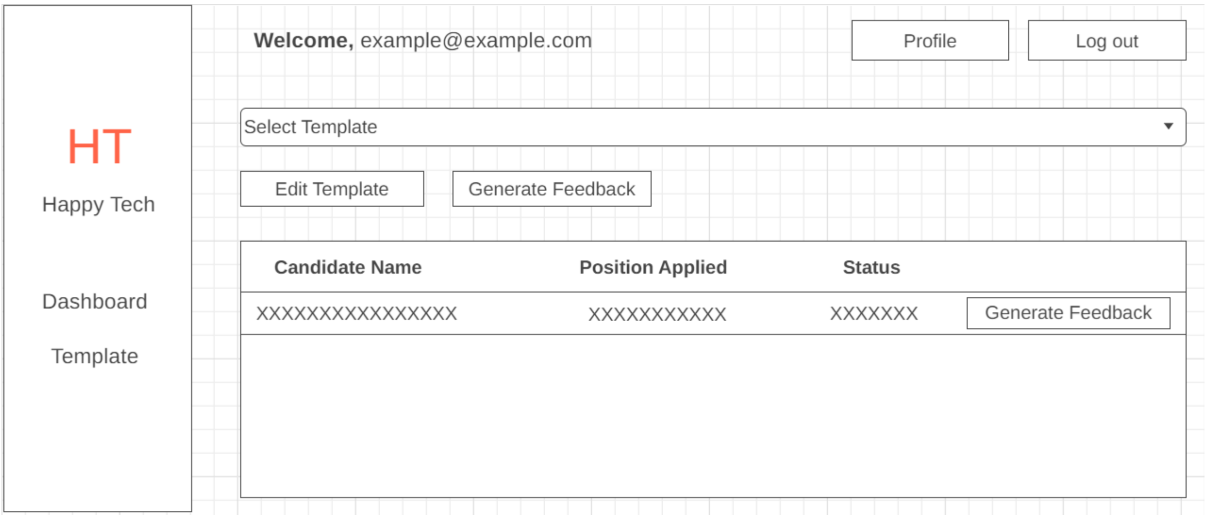


Figure 4 Login & Registration Wireframe

The users who want to give feedback need to login into the application. The existing users can make use of the login feature whereas the new users need to register using the Registration Form.

Reviewer Dashboard

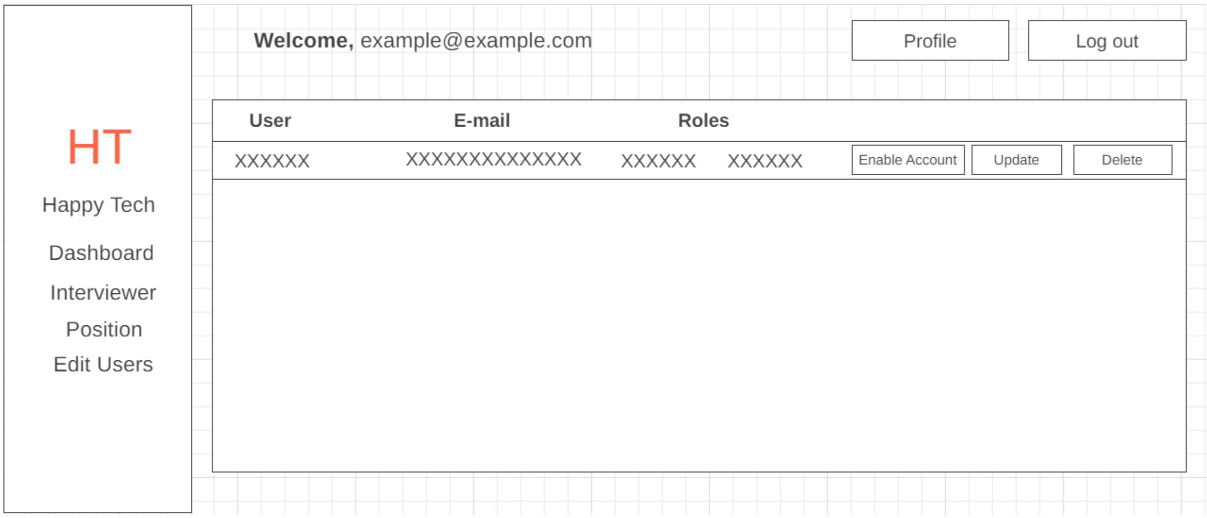


Candidate Name	Position Applied	Status	
XXXXXXXXXXXXXXXX	XXXXXXXXXXX	XXXXXXX	Generate Feedback

Figure 5 Reviewer Dashboard

The users with the reviewer role will be redirected to the dashboard after successful login or registration, where the main application features are accessible. In the Dashboard, reviewers can create and edit templates and can even access the list of candidates applied to various positions listed on the homepage.

Admin Dashboard



User	E-mail	Roles				
XXXXXX	XXXXXXXXXXXXXX	XXXXXX	XXXXXX	Enable Account	Update	Delete

Figure 6 Admin Dashboard

The users with the admin role will be redirected to the admin dashboard where he/she can access a list of users registered with the application. The user with the admin role can able to perform actions on the user profiles.

Create Template

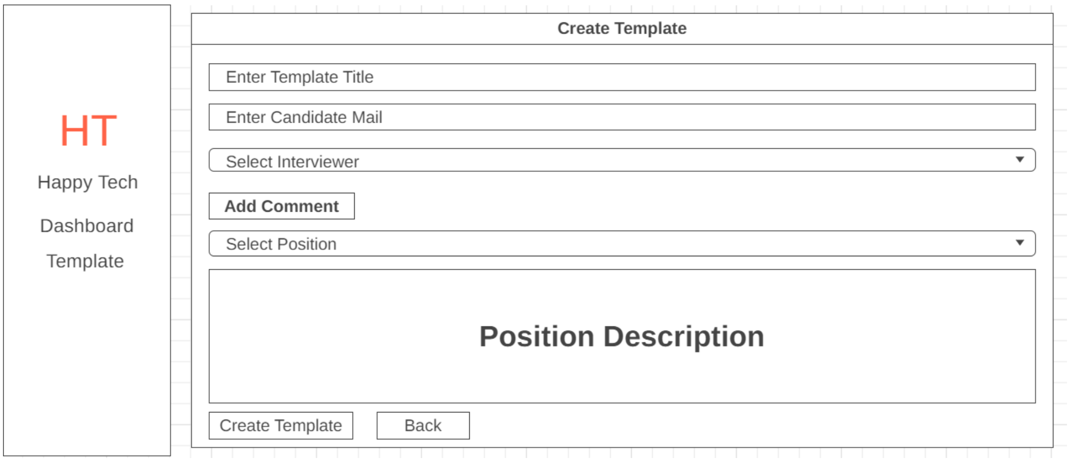


Figure 7 Create Template Wireframe

A template is used to create feedback for the candidate. So, the create template page consists of a form with fields like a name for the template, button to add feedbacks, positions, interviewers and the candidate mail ID field.

Edit and Delete Template

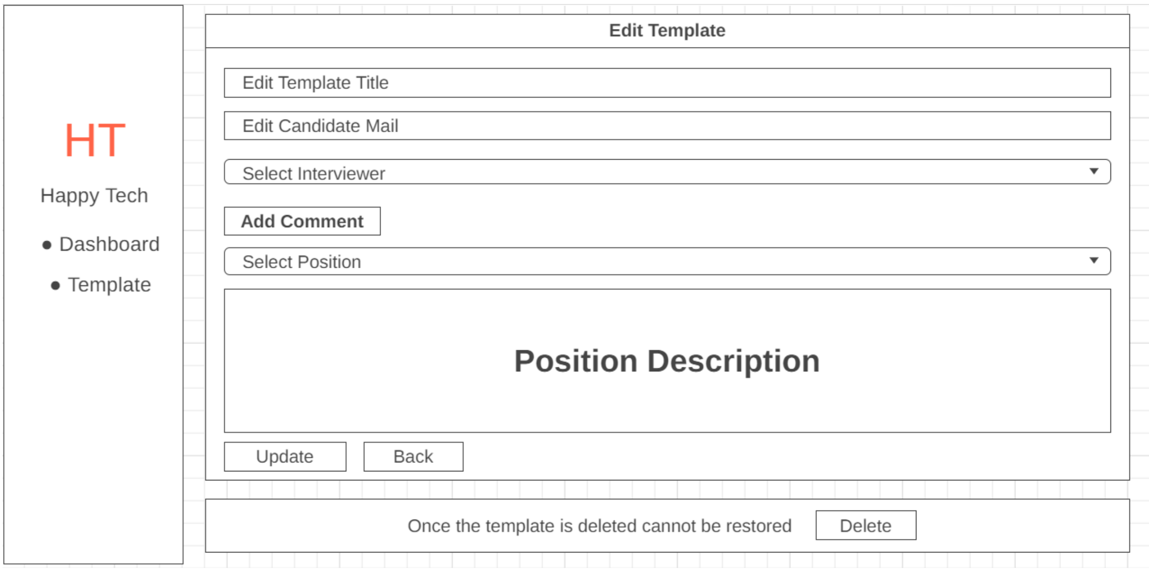


Figure 8 Edit & Delete Template Wireframe

Edit Template page will allow the reviewer to update all existing values or delete the entire template. Information like template name, position, candidate mail, and also existing feedback comments linked with the selected template.

Create Position, Interviewer, and Edit Profile

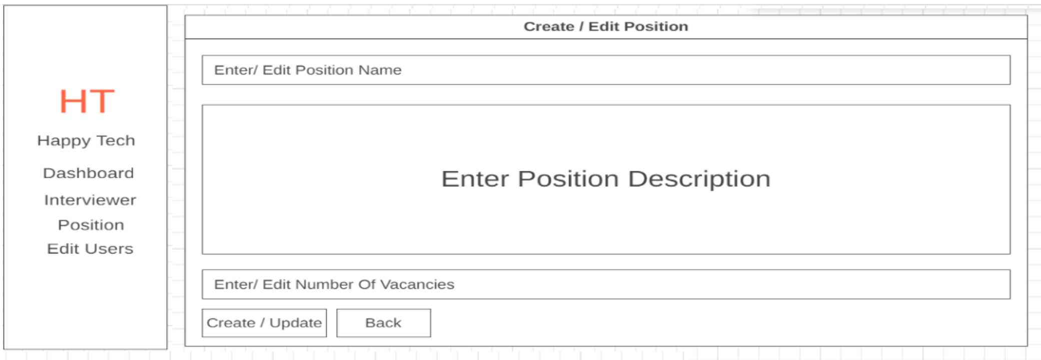
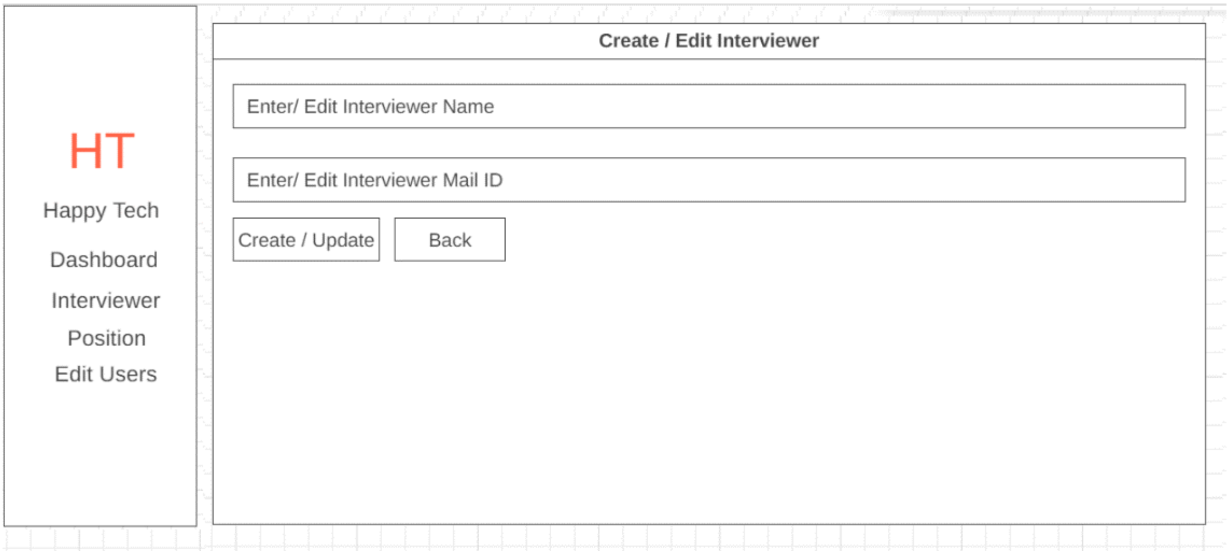


Figure 9 Create/Edit Position Wireframe


Create / Edit Interviewer
HT
Happy Tech
Dashboard
Interviewer
Position
Edit Users
Enter/ Edit Interviewer Name
Enter/ Edit Interviewer Mail ID
Create / Update
Back

This application also has extra features such as creating or updating the existing positions, interviewers. These features are available only to the users with the admin role and can be accessed from the application menu.

Figure 10 Create/Edit Interviewer Wireframe

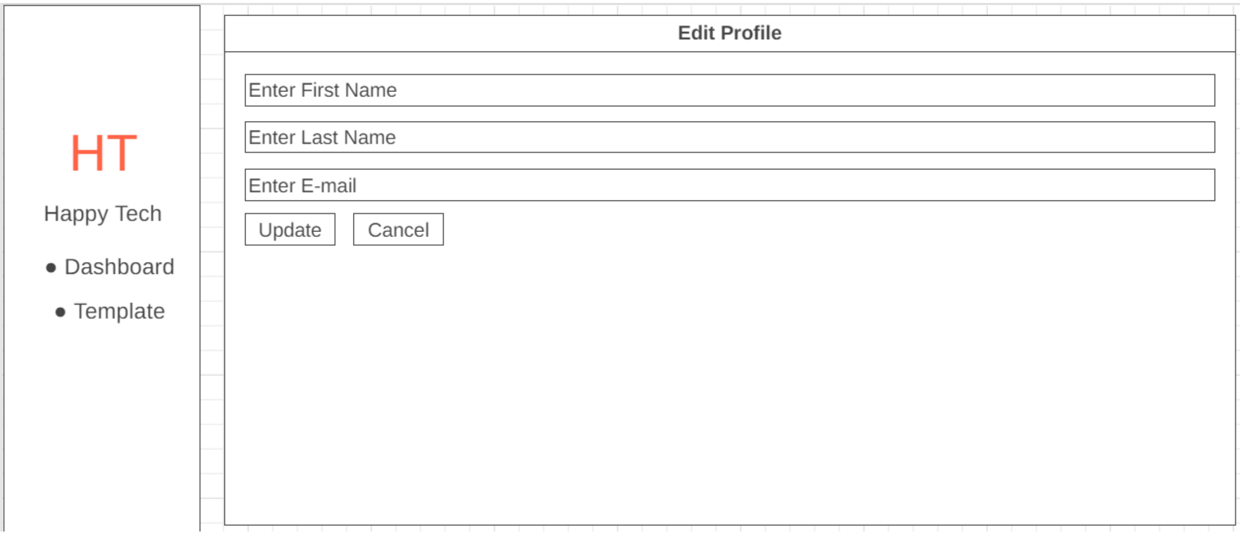


Figure 11 Edit Profile Wireframe

The "Edit Profile" feature enables users to update the details linked with their user account. Details like first name, last name, and E-Mail address can be updated with this feature. This feature is accessible to every user on the top-right side of the dashboard page.

Generate Feedback

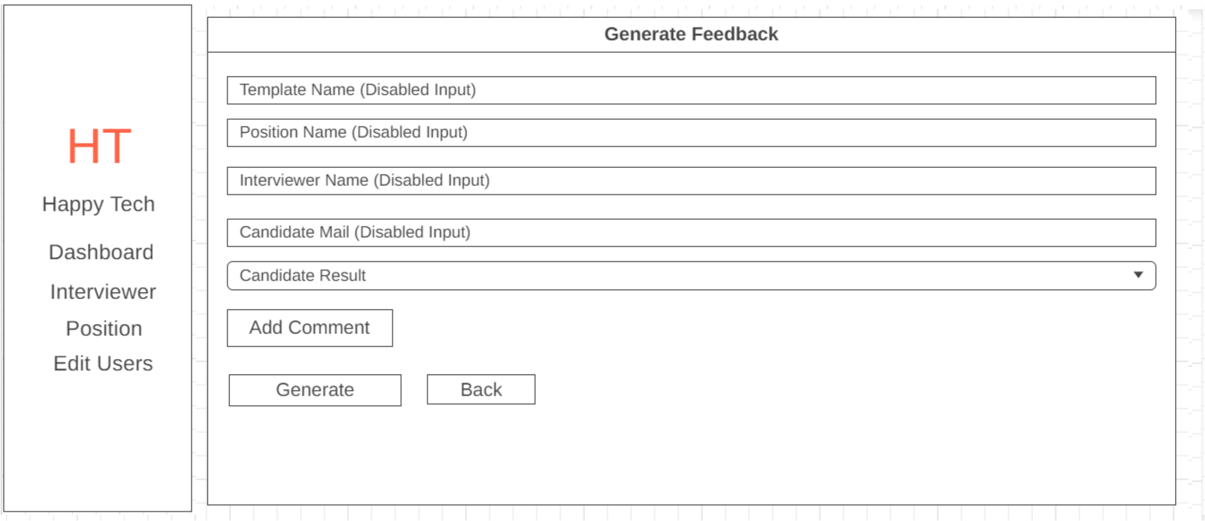

HT
Happy Tech
Dashboard
Interviewer
Position
Edit Users
Generate Feedback
Template Name (Disabled Input)
Position Name (Disabled Input)
Interviewer Name (Disabled Input)
Candidate Mail (Disabled Input)
Candidate Result
Add Comment
Generate
Back

Figure 12 Generate Feedback Wireframe

Generate Feedback page will let the reviewers give feedback on the candidate application. This feature allows reviewers to add extra comments if needed. And the email will be sent to candidates once the process of giving feedback is completed for all candidates.

Database design

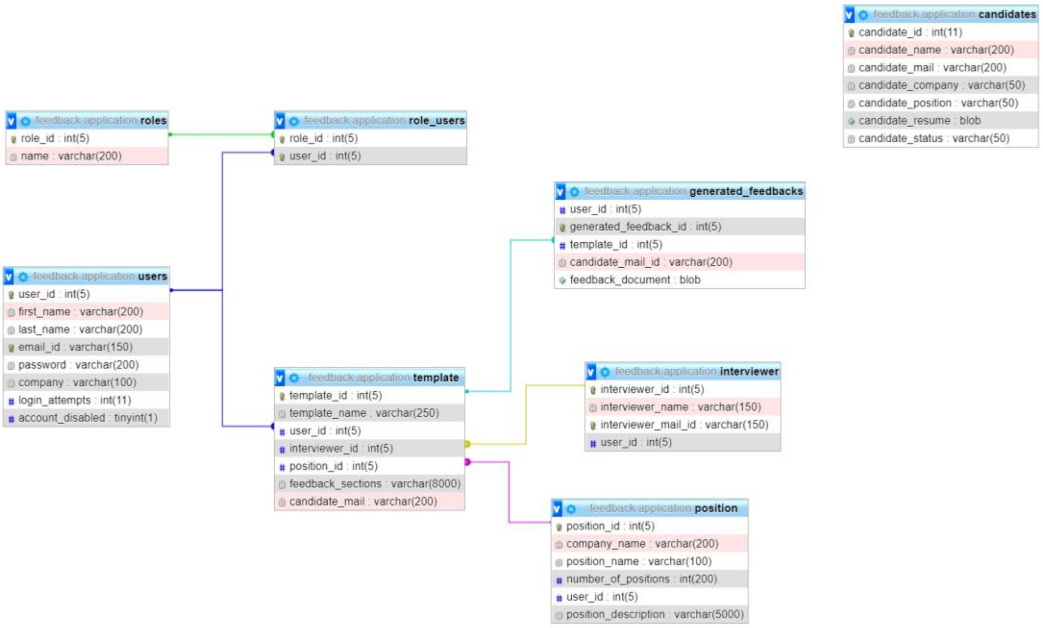

feedback application candidates
candidate_id : int(11)
candidate_name : varchar(200)
candidate_mail : varchar(200)
candidate_company : varchar(50)
candidate_position : varchar(50)
candidate_resume : blob
candidate_status : varchar(50)
feedback application roles
role_id : int(5)
name : varchar(200)
feedback application role_users
role_id : int(5)
user_id : int(5)
feedback application generated_feedbacks
user_id : int(5)
generated_feedback_id : int(5)
template_id : int(5)
candidate_mail_id : varchar(200)
feedback_document : blob
feedback application users
user_id : int(5)
first_name : varchar(200)
last_name : varchar(200)
email_id : varchar(150)
password : varchar(200)
company : varchar(100)
login_attempts : int(11)
account_disabled : tinyint(1)
feedback application template
template_id : int(5)
template_name : varchar(250)
user_id : int(5)
interviewer_id : int(5)
position_id : int(5)
feedback_sections : varchar(8000)
candidate_mail : varchar(200)
feedback application interviewer
interviewer_id : int(5)
interviewer_name : varchar(150)
interviewer_mail_id : varchar(150)
user_id : int(5)
feedback application position
position_id : int(5)
company_name : varchar(200)
position_name : varchar(100)
number_of_positions : int(200)
user_id : int(5)
position_description : varchar(5000)

Figure 13 Database Design

In this project, the MySQL database is used to store the data. And the application database consists of different tables where each table is designed to store the different data.

Users

The 'users' table has a one-to-many relationship with table 'Template' in the database. The users table stores the information about the reviewers and admins using the application.

Descriptive Name	Description	Type*	Column Name	Valid Values	Index Column	Allow Nulls
User ID	It is an auto-increment field and the value is added automatically when the user is created	integer	user_id		Primary Index	NO
First Name	This field stores the first name of the user.	varchar(200)	first_name	Any string without special characters or numbers		NO

Last Name	This field stores the last name of the user.	varchar(200)	last_name	Any string without special characters or numbers		NO
E-Mail ID	This field stores the valid e-mail id of a user.	varchar(150)	email_id	Valid E-Mail Address	Unique Index	NO
Password	This field stores the encrypted password of the user account.	varchar(200)	password	Any string containing characters, numbers, and alphabets		NO
Company	This field stores the information about the user's company. By default, the user company is 'Happy Tech'.	varchar(100)	company			NO
Failed Login Attempts	This value stored in this field is reduced by one every time the user tries to log in with the wrong password. By default, the number of login attempts given is 3.	integer	login_attempts			NO
Account Disabled	This field stores the status of a user account. By default value is set to false, If the login attempts value equals zero, then the value of this field will be set to true.	tinyint	account_disabled			NO

Template

The 'template' table store the templates created by the reviewers. This table has a one-to-many relationship with the 'generated_feedback' table.

Descriptive Name	Description	Column Name	Type*	Valid Values	Index Column	Allow Nulls
Template ID	This field stores the id of the template that is unique and auto-incremented when a new record is created.	template_id	integer		Primary Index	NO
Template Name	This field stores the template name.	template_name	varchar(250)	Any String value		NO
User ID	This field stores the IDs of the users that exist in the user table.	user_id	integer	valid user ID		NO
Interviewer ID	This field stores the IDs of the interviewers who exist in the interviewer table.	interviewer_id	integer	valid interviewer ID		NO
Position ID	This field stores IDs of the positions that exist in the position table.	position_id	integer	valid position ID		NO
Feedback Comments	This field stores the information about the comments added as feedback in array format.	feedback_sections	varchar(8000)			
Candidate Mail	This field stores the mail ID of a candidate for who the feedback is for.	candidate_mail	varchar(200)	valid candidate mails		NO

Position

The ‘position’ table is designed to store the details about the vacancies in a company.

Descriptive Name	Description	Column Name	Type*	Valid Values	Index Column	Allow Nulls

Position ID	This field stores the id of the position that is unique and auto-incremented when a new record is created.	position_id	integer		Primary Index	NO
Company Name	This field stores the company in which the position exists. By default, the value is ‘Happy Tech’.	company_name	varchar(200)	Stores the value of default company		NO
Number of Positions	This field stores the number of vacancies available for a specific position in the company.	number_of_positions	int	Stores the vacancy number of default company		NO
User ID	This field stores the IDs of the users that exist in the user table.	user_id	int	valid user ID		NO
Position Name	This field stores the name of the position.	position_name	varchar(200)	stores any string		NO
Position Description	This field stores the description of the vacancy available in the company.	position_description	varchar(5000)	stores any string		NO

Interviewer

The ‘interviewer’ table is designed to store the details about the employees in a company who takes an interview.

Descriptive Name	Description	Column Name	Type*	Valid Values	Index Column	Allow Nulls
Interviewer ID	This field stores the id of the interviewer that is unique and auto-incremented when a new record is created.	interviewer_id	integer		Primary Index	NO
Interviewer Name	This field stores the interviewer's name. By default, the interviewer's name will be fetched from the reviewer's first name	interviewer_name	varchar(150)			NO

	and last name while creating the account.					
Interviewer Mail ID	This field stores the interviewer's E-Mail ID. By default, the interviewer's name will be fetched from the reviewer's first name and last name while creating the account.	interviewer_mail_id	varchar(150)	stores valid interviewer mail ID	Unique Index	NO
User ID	This field stores the IDs of the users that exist in the user table.	user_id	integer	stores the valid user ID		NO

Generated Feedbacks

The ‘generated_feedbacks’ table store the data about the feedback given to the candidates.

Descriptive Name	**Description**	**Column Name**	**Type***	**Valid Values**	**Index Column**	**Allow Nulls**
User ID	This field stores the IDs of the users that exist in the user table.	user_id	integer	stores the valid user ID		NO
Generated Feedback ID	This field stores the id of the generated feedback that is unique and auto-incremented when a new record is created.	generated_feedback_id	integer		Primary Index	NO
Feedback Document	This field stores the pdf format of the feedback given by the reviewer on the candidate application.	feedback_document	longbob			NO
Position ID	This field stores IDs of the positions that exist in the position table.	position_id	integer	stores valid position ID		NO

Candidate Mail ID	This field stores the mail ID of a candidate for who the feedback is for.	candidate_mail_id	varchar(200)	stores valid candidate mail ID		NO

Candidates

The ‘candidates’ table deals with the information about the candidates who applied for the vacancies in the applications.

Descriptive Name	Description	Column Name	Type*	Valid Values	Index Column	Allow Nulls
Candidate ID	This field stores the id of the candidate that is unique and auto-incremented when a new record is created.	candidate_id	integer		Primary Index	NO
Candidate Name	This field stores the candidate's name.	candidate_name	varchar(200)			NO
Candidate Mail	This field stores the candidate's E-Mail ID.	candidate_mail	varchar(200)	stores valid candidate mail ID		NO
Candidate Company	This field stores the company for which the candidate is applied for.	candidate_company	varchar(50)	Stores the value of default company		NO
Candidate Position	This field stores the specific position in a company for which the candidate is applied for.	candidate_position	integer	stores valid position ID		NO
Candidate Resume	This field stores the pdf or word format of the candidate's resume.	candidate_resume	longblob			NO
Candidate Status	This field stores the status of the candidate application. By default, the value is ‘Pending’.	candidate_status	varchar(50)	stores the default value of the application status		NO

Roles

The ‘roles’ table is designed to store the details about the employee roles in a company that uses the application.

Descriptive Name	Description	Column Name	Type*	Valid Values	Index Column	Allow Nulls
Role ID	This field stores the id of the role that is unique and auto-incremented when a new record is created.	role_id	integer		Primary Index	NO
Name	This field stores the Role name.	name	varchar(200)			NO

Role Users

The ‘role_users’ table is designed to store the details about the employee and their roles in a company.

Descriptive Name	Description	Column Name	Type*	Valid Values	Index Column	Allow Nulls
Role ID	This field stores IDs of the roles that exist in the role table.	role_id	integer	stores valid role ID		NO
User ID	This field stores the IDs of the users that exist in the user table.	user_id	integer	stores the valid user ID		NO

Software Design

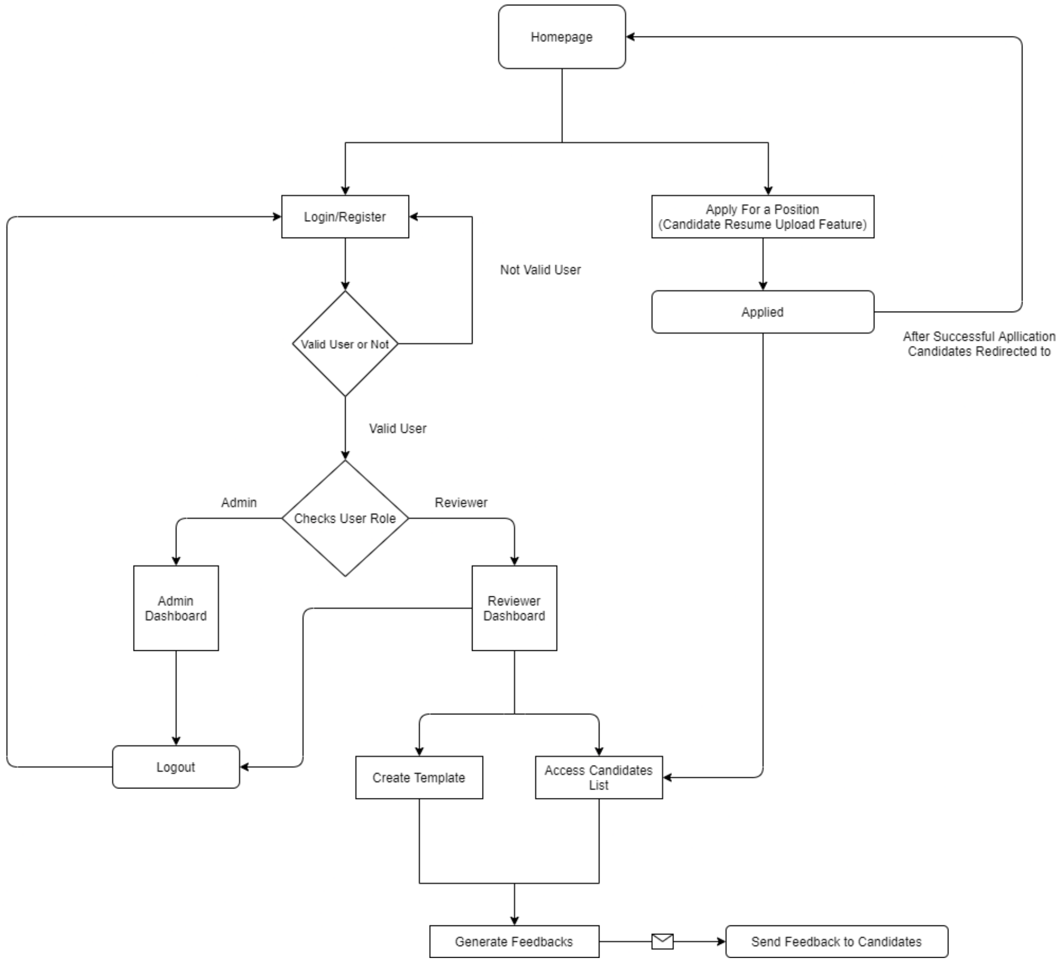


Figure 14 Flowchart Of An Application

The application consists of three types of users: admins, reviewers, and candidates. Each role is designed with a set of permissions. For example, the users with an admin role can able view the list of reviewers whereas the reviewer able to see the list of candidates. And each role is intended to do a set of actions like generating feedback for candidates using the reviewer role and unlocking or deleting the reviewer account using the admin role.

Candidates

To apply for jobs or any other services like CV review, candidates can make use of the form on the landing page. The form will show the list of services or positions that are vacant in the company. Once the candidate completes the application form, he/she can view the update on their application using the 'Candidate Status' functionality.

Reviewers

The main aim of the reviewer role in this application is to give feedback on the submitted applications. So, the reviewer can able to see the list of candidates who applied for a job vacancy. Reviewers need to create templates to give feedback on the candidate application. So, the templates help the reviewer to give the same feedback for more than one candidate without writing the entire feedback. And once the vacancies of any available positions are filled, then the application will send the feedback mail for all applied candidates.

Admins

Admins in this application are the users with the highest role permissions. These users will have the ability to change the permissions of reviewers or even other admins like deleting or updating the account and even have the ability to unlock the locked accounts.

Software Development Documentation

Each member of our team contributed their efforts in developing the web application. My contribution to this application contains a list of few user stories. These user stories are explained in detail below.

Login and Registration

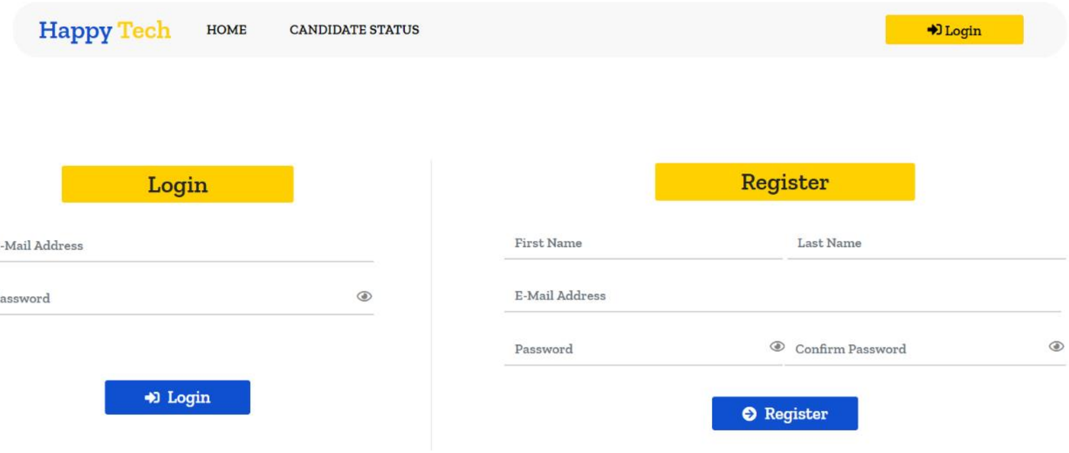
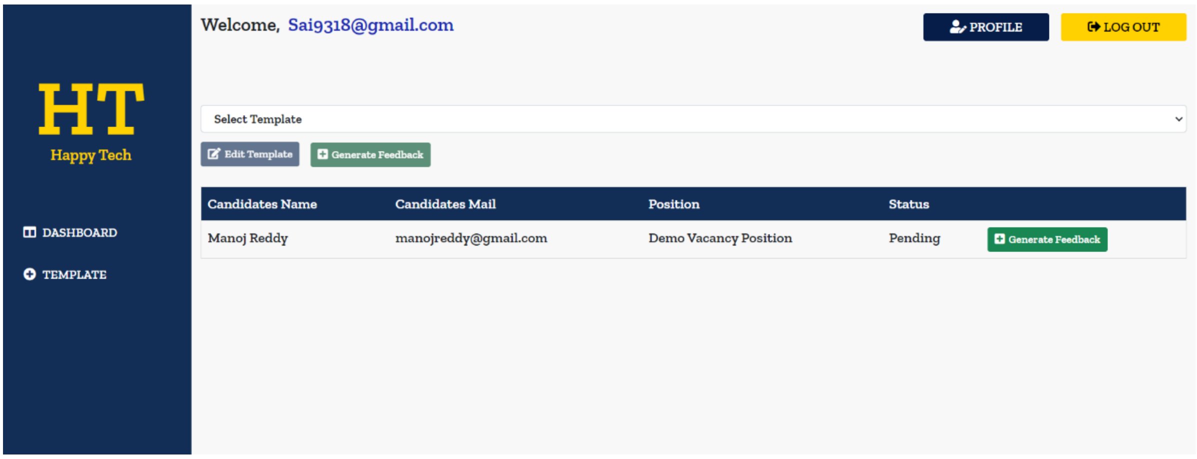


Figure 15 Login And Registration Page

The Login and Registration features are for reviewing and administration purposes. Once the reviewer is registered, he/she can access the main functionalities of this web application. While registering as a user, the reviewers must specify their details like first name, last name, mail id, password. Once registered, the reviewers can log in to their account using mail id and password. Admins and reviewers can use the same login page to access their accounts. The login functionality use ‘Account Lockout’ concept to keep the account secure from automated attacks like brute-force attack. This feature automatically locks the user account after a certain number of invalid attempts and won’t allow the attacker or the valid user to login into the account even if they provide valid credentials. These locked/disabled accounts can be enabled by the admin with the ‘Enable Account’ feature in the admin dashboard.

Admin Dashboard and Reviewer Dashboard



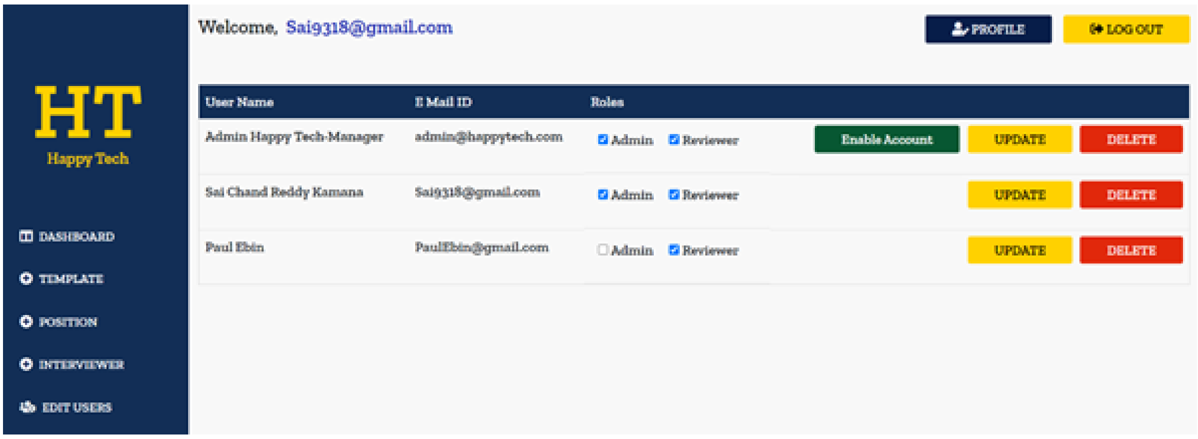
Welcome, Sai9318@gmail.com

Select Template

Candidates Name	Candidates Mail	Position	Status	
Manoj Reddy	manojreddy@gmail.com	Demo Vacancy Position	Pending	Generate Feedback

Figure 16 Reviewer Dashboard

On successful login, the user with the reviewer role is redirected to the 'Dashboard' page where they can access all functionalities available in the application. On the Reviewer Dashboard page, reviewers can work with features like creating, editing, and deleting the templates and generating feedbacks. Reviewers can start generating the feedback for candidates using 'Generate Feedback' in the candidate's section or the 'Generate Feedback' option in the template section. The feedback feature in the candidate section will automatically create a default template with the details linked with the candidate profile to generate feedback.



Welcome, Sai9318@gmail.com

User Name	E Mail ID	Roles			
Admin Happy Tech-Manager	admin@happytech.com	☑ Admin ☑ Reviewer	Enable Account	UPDATE	DELETE
Sai Chand Reddy Kamana	Sai9318@gmail.com	☑ Admin ☑ Reviewer		UPDATE	DELETE
Paul Ebin	PaulEbin@gmail.com	□ Admin ☑ Reviewer		UPDATE	DELETE

Figure 17 Admin Dashboard

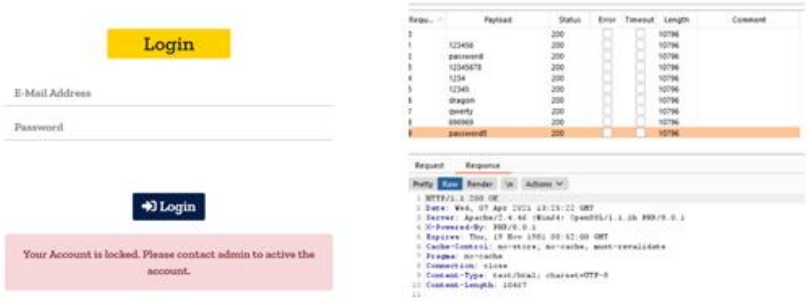
And the user with the admin role is redirected to the admin dashboard. In this dashboard, admins can access a list of users registered in the application. And admin has the ability to unlocking the reviewer's account, edit the reviewer's role, or deleting the reviewer's account from the system.

Testing

Few testing techniques are implemented to fix bugs in the application. These testing techniques include Brute force attacks and Automated testing.

Brute Force Attack - Broken Authentication

In a broken authentication attack, the hackers can log in to the legitimate user account using techniques like session hijacking, automated attacks, and brute force attacks. Few techniques are implemented to avoid this vulnerability.



Your Account is locked. Please contact admin to active the account.

Requ...	Payload	Status	Error	Timeout	Length	Comment
0		200	□	□	10796	
1	123456	200	□	□	10796	
2	password	200	□	□	10796	
3	12345678	200	□	□	10796	
4	1234	200	□	□	10796	
5	12345	200	□	□	10796	
6	dragon	200	□	□	10796	
7	qwerty	200	□	□	10796	
8	696969	200	□	□	10796	
9	password5	200	□	□	10796	

```
HTTP/1.1 200 OK
Date: Wed, 07 Apr 2021 13:15:22 GMT
Server: Apache/2.4.46 (Win64) OpenSSL/1.1.1h PHP/8.0.1
X-Powered-By: PHP/8.0.1
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 10467
```

Figure 17 Brute Force Attack

As you can see in the above figure, the brute-force attack pushed the account into lock mode. And the login functionality didn't allow the user to log into their account even when a valid password is provided.

Selenium - Automated Testing

Testing with Selenium WebDriver is also performed on the feedback application using few test cases shown below.

Feature	Scenario	Expected	Output	Result	Chrome	Edge	Firefox
Login	Correct E-Mail & Password	Successful Login	Successful Login	Pass	Passed	Passed	Passed

	Correct E-Mail & Incorrect Password	Failed Login	Incorrect E-Mail or Password	Pass	Passed	Passed	Passed
	Incorrect E-Mail & Correct Password	Failed Login	Incorrect E-Mail or Password	Pass	Passed	Passed	Passed
	Incorrect E-Mail & Password	Failed Login	Incorrect E-Mail or Password	Pass	Passed	Passed	Passed
	Correct E-Mail & Empty Password	Asks for Input	Required Field	Pass	Passed	Passed	Passed
	Empty E-Mail & Correct Password	Asks for Input	Required Field	Pass	Passed	Passed	Passed
	Empty E-Mail & Password	Asks for Input	Required Field	Pass	Passed	Passed	Passed
Register	All Empty Input Fields	Asks for Input	Required Field	Pass	Passed	Passed	Passed
	Specific Empty Fields	Asks for Specific Input	Required Specific Field	Pass	Passed	Passed	Passed
	Any incorrect characters in the E-Mail field	Asks to check for Input	Check provided Details	Pass	Passed	Passed	Passed
	All Valid Inputs	Successful Registration	Successful Registration	Pass	Passed	Passed	Passed
Dashboard Menu	Valid Page URL Redirection	Successful Redirect	Successful Redirect	Pass	Passed	Passed	Passed
	Invalid Page URL Redirection	Redirects to Dashboard or valid page	Redirects to Dashboard or valid page	Pass	Passed	Passed	Passed

Homepage Menu	Valid Page URL Redirection	Successful Redirect	Successful Redirect	Pass	Passed	Passed	Passed
	Invalid Page URL Redirection	Redirects to Homepage or valid page	Redirects to Homepage or valid page	Pass	Passed	Passed	Passed

Conclusion

Developing the feedback application helped me to learn few concepts like MVC, designing software, database, and wireframe, how to work with the web framework and how to prevent vulnerabilities using some popular techniques.

The developed application can be improved with other features like preview resumes of applied candidates, generating feedbacks for multiple candidates at a time, listing the available vacancies in the application in a clear and visible format. And allowing candidates to create accounts to keep track of their newly applied jobs and also to preview and edit their resumes.

References

Google.com. (2017). Google Chrome Web Browser. [online] Available at: https://www.google.com/intl/en_uk/chrome/.

Apache Friends (2019). XAMPP Installers and Downloads for Apache Friends. [online] Apachefriends.org. Available at: https://www.apachefriends.org/index.html.

Atom. (2019). A hackable text editor for the 21st Century. [online] Available at: https://atom.io/docs.

Duckett, Jon. HTML & CSS : Design and Build Websites . Indianapolis, Ind: Wiley, 2011. Print.

After Interview 2018. After Interview App - Must-have tool for your job search!. [online] Afterinterview.com. Available at: <https://afterinterview.com/> [Accessed 12 February 2021].

Recruiterbox 2011. Collect Interview Feedback | Recruiterbox. [online] Available at: <https://recruiterbox.com/recruitment-software-features/collect-interview-feedback> [Accessed 12 February 2021].

Codeigniter 2019. Tutorial — CodeIgniter 3.1.11 documentation. [online] Available at: <https://codeigniter.com/userguide3/tutorial/index.html> [Accessed 02 February 2021].

Niederst Robbins, Jennifer. Learning Web Design a Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. Fifth edition. Sebastopol, CA: O'Reilly, 2018. Print.

Carr, David, and Markus Gray. Beginning PHP. 1st ed. Birmingham: Packt Publishing, Limited, 20.

Wibawa, S C et al. "Online Test Application Development Using Framework CodeIgniter." IOP Conference Series. Materials Science and Engineering. Vol. 296. IOP Publishing, 2018. 12041–. Web.18. Print.