

Right Resource Fit

1. Title: Right Resource Fit

2. Project Statement:

Right Resource Fit: This project will enable the Human Resource Management Team to handle HR functions related to recruitment.

Outcomes:

1. Users can register as Applicants or HR.
2. Users can log in to the system.
3. Users can reset their passwords.
4. HR can create, update, and delete job postings.
5. Applicants can search for jobs based on location, skills, etc.
6. Applicants can apply for jobs by filling out a job application form.
7. Applicants can view the status of their job applications.
8. HR can track applicants and their application statuses.

3. Modules to be Implemented:

1. Module A: User Management
2. Module B: Job Posting and Application Management
3. Module C: Interaction and Tracking
4. Module D: Reporting and Administration

4. Week-wise Module Implementation and High-level Requirements with Output

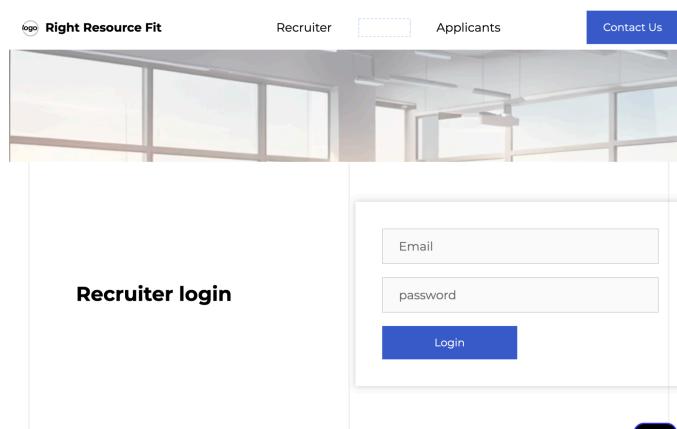
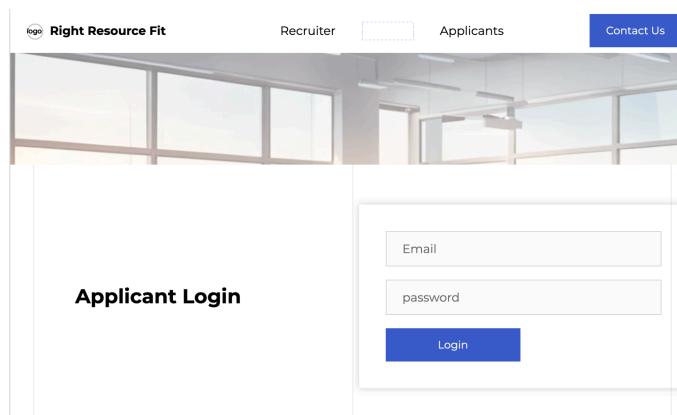
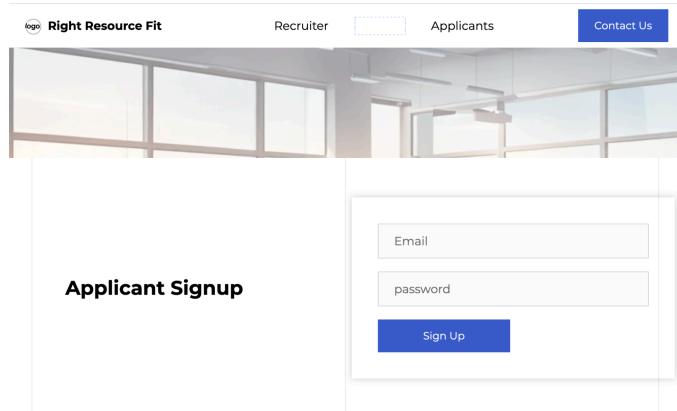
Milestone 1: Weeks 1-2

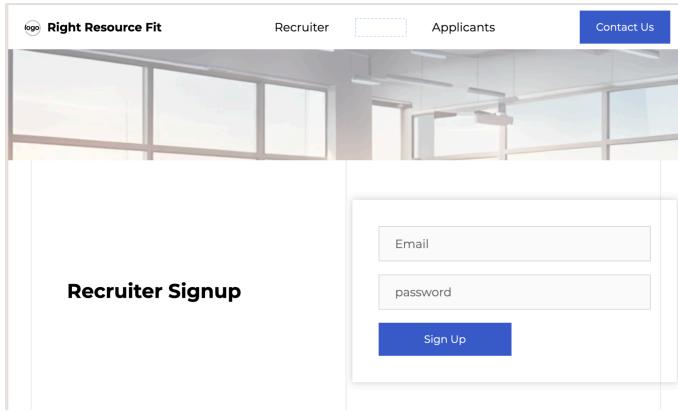
Module A: User Management

- Implement basic site structure with redirection to /home with dummy data.
- Implement static /about route.
- Implement login, registration, and password reset flows with fail scenarios.
- Setup database and backend integration to support all functionalities.
- Setup templates for Job Posting, Application Management, Interaction, and Reporting.
- Ensure the application is functional with basic navigation.

Output :

- User registration, login, and password reset forms





- Home page with registration and login options

Milestone 2: Weeks 3-4

Module B: Job Posting and Application Management

- Implement job posting creation, update, and deletion functionality for HR.
- Allow applicants to search for jobs by location, skills, etc.
- Setup database schema for storing job postings and applications.
- Integrate frontend with backend to support job posting and application management functionalities.

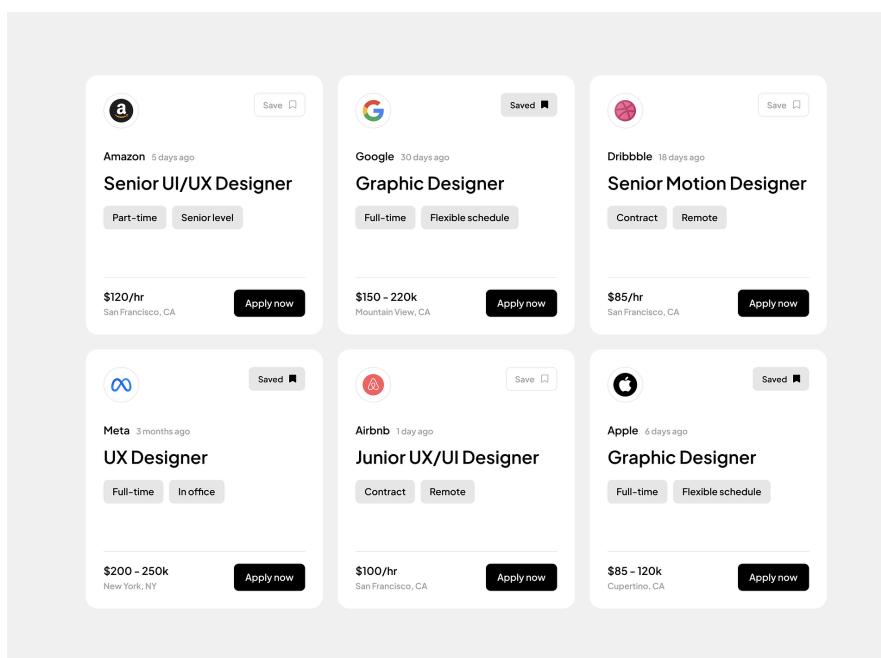
Output :

- Job posting creation and management forms

This image shows a detailed view of a job posting creation form. It includes fields for job title, description, employment type, working schedule, salary, and hiring multiple candidates.

Job title A job title must describe one position only	<input kitchen="" staff\""="" type="text" value="e.g. \"/>
Job description Provide a short description about the job. Keep it short and to the point.	<input style="height: 100px; margin-bottom: 5px;" type="text"/> <div style="display: flex; justify-content: space-between;"> B I U S D E 200 words </div>
Employment type Description text goes in ehre	<input checked="" type="checkbox"/> Full-time <input checked="" type="checkbox"/> Part-time <input type="checkbox"/> On demand <input type="checkbox"/> Negotiable
Working schedule You can pick multiple work schedules.	<input type="button" value="Pick working schedule"/> <input type="radio"/> Day shift <input type="radio"/> Night shift <input type="radio"/> Weekend availability
Salary Choose how you prefer to pay for this job.	<input type="radio"/> Hourly <input checked="" type="radio"/> Custom Amount you want to pay: <input type="text" value="35,000"/> How you want to pay: <input type="button" value="Yearly"/> <input type="checkbox"/> Salary is negotiable
Hiring multiple candidates? This will be displayed on job page for candidates to see.	<input type="checkbox"/> Yes, I am hiring multiple candidates

- Job search interface



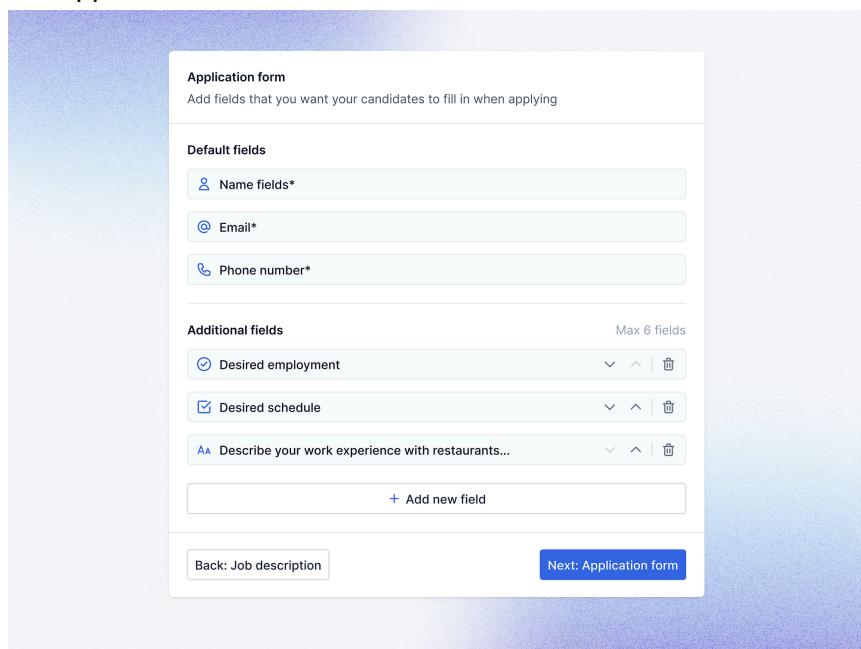
Milestone 3: Weeks 5-6

Module C: Interaction and Tracking

- Implement job application form for applicants.
- Allow HR to track applicants and their application statuses.
- Implement CRUD operations for job applications.
- Setup notification system for application status updates.
- Integrate interaction features with existing job posting and application management functionalities.

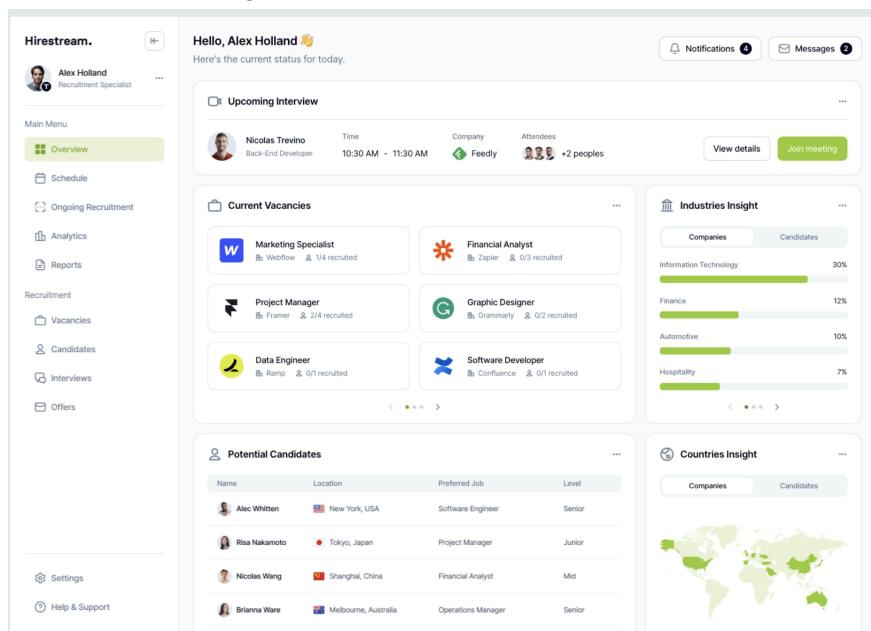
Output :

- Job application form



The screenshot shows a 'Job application form' configuration page. At the top, it says 'Application form' and 'Add fields that you want your candidates to fill in when applying'. Below this, under 'Default fields', there are three required fields: 'Name fields*', 'Email*', and 'Phone number*'. Under 'Additional fields', there are three optional fields: 'Desired employment' (checked), 'Desired schedule' (checked), and 'Describe your work experience with restaurants...'. A button '+ Add new field' is available. At the bottom, there are 'Back: Job description' and 'Next: Application form' buttons.

- Application tracking interface for HR



Milestone 4: Weeks 7-8

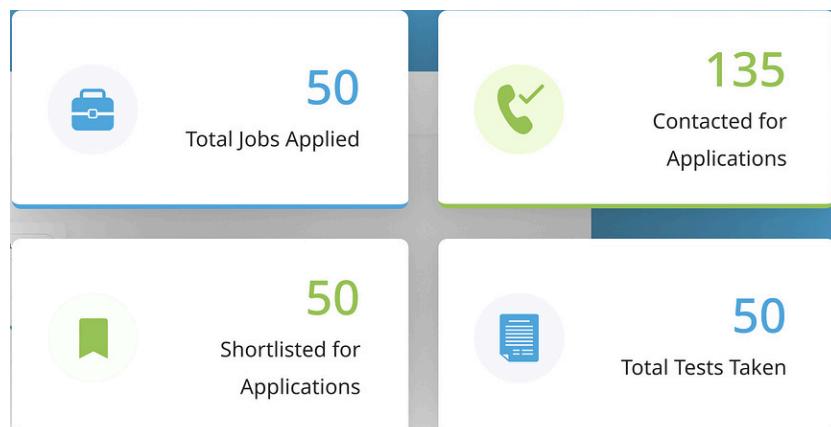
Module D: Reporting and Administration

- Implement reporting functionalities for application status and job postings.
- Develop interfaces for generating and viewing reports.
- Implement user and role management functionalities for admins.
- Allow HR to update application statuses and notify applicants.
- Integrate reporting and administrative features with the overall application.

Output :

- Reporting interface

The screenshot displays a reporting interface for a job posting. At the top, there's a header for 'Unity Developer' located in 'San Francisco, CA'. The job details include 'Full time', 'Work From Home', '\$ 250k - 300k / year', and 'Posted On 24/11/2023'. Below this, a summary bar shows 'No of Applicants for the Job: 21', 'Last active: 23/01/2023', and 'Status: Closed'. The main section is titled 'Applicants List' and contains four cards, each representing an applicant named 'John David' (johndavid@gmail.com). Each card includes a profile picture, the application date ('Applied on Oct 26, 2023'), and a list of skills: UI, UX, Website Design, Mobile Design, Design Wireframes, Webflow, and 8+. Each card also has three buttons: 'View Profile', 'Download CV', and 'Watch video'.



- User and role management interface

Member Invitation (2)				
All user role	NAME	EMAIL	ROLE	2FA STATUS
 Bagus Fikri You	BagusFikri@gmail.com		Admin (Owner)	✓ Enabled
 Constantin Raihan	Constantin93@gmail.com		Payer	✗ Disabled
 Salma Waheed	Salma95@hotmail.com		Preparer	✓ Enabled
 Trantow Rae Jr.	Rae.Trantow@gmail.com		Preparer	✗ Disabled
 Susan Jones	Susan.Jones@yahoo.com		Employee	✗ Disabled
 Sydnie Stark	Sydnie_Stark@hotmail.com		Employee	✓ Enabled
 Rashad J. Shields	Rashad_Shields@yahoo.com		Employee	✗ Disabled
 Daron Nader	Daron.Nader@hotmail.com		Employee	✓ Enabled
 Jennifer Dortha	Dortha_Feest98@gmail.com		Viewer	✓ Enabled
 Carolina Agusto	Carolina27@gmail.com		Viewer	✓ Enabled
Load More				

Evaluation Criteria:

Milestone 1 Evaluation (Week 2):

- Completion of user authentication and registration.
- Completion of login after authorization.
- Implementation of the static /about route.
- Code checked into version control.

Milestone 2 Evaluation (Week 4):

- Completion of job posting creation and management.
- Job search functionality.
- Integration of frontend with backend for job posting and search.
- Code checked into version control.

Milestone 3 Evaluation (Week 6):

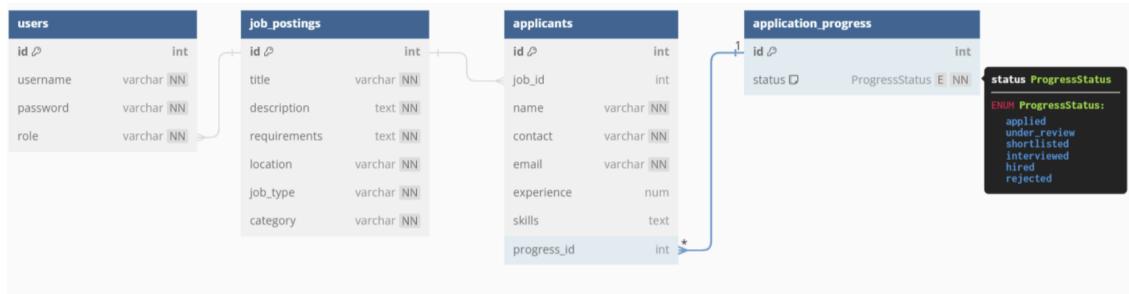
- Implementation of job application form.
- Tracking of applicants and application statuses by HR.
- Integration of interaction features with job posting and application management.
- Code checked into version control.

Milestone 4 Evaluation (Week 8):

- Implementation of reporting and user management functionalities.
- HR capabilities to update application statuses and notify applicants.
- Integration of reporting and administrative features.
- Tagging code in version control.

5. Design Diagrams:

Design diagrams (/flow charts/DB design etc)



6. Class Diagram :

auth/ Authentication Service

Service Description: Handles user authentication processes including registration, login, password reset, and confirmation.

Endpoints:

- *POST /login*
 - Payload: { email: "user@example.com", password: "password" }
 - Responses: 200, 401, 404
- *POST /register*
 - Payload: { name: "User", email: "user@example.com", password: "password", role: "applicant" }
 - Responses: 201, 400, 409
- *POST /forgot*
 - Payload: { email: "user@example.com" }
 - Responses: 200, 404
- *GET /reset/<token>*
 - Query Param: newpassword, token
 - Responses: 200, 404
- *GET /confirm/<token>*
 - Query Param: token
 - Responses: 200, 400, 401

job/ Job Posting and Application Management Service (protected)

Service Description: Handles job postings and applications, allowing HR to manage job listings and applicants to search and apply for jobs.

Endpoints:

- *POST /postings*
 - Payload: { title: "New Job", description: "Job details", location: "City", skills: ["skill1", "skill2"], postedBy: "hr@example.com" }
 - Responses: 201, 400, 404
- *PUT /postings/<id>*

- Payload: { title: "Updated Job", description: "Updated details", location: "City", skills: ["skill1", "skill2"] }
 - Responses: 204, 401, 404
- *DELETE /postings/<id>*
 - Query Param: jobID
 - Responses: 200, 404
- *GET /postings*
 - Query Param: location, skills
 - Responses: 200, 404
- *POST /applications*
 - Payload: { jobID: "123", applicantID: "applicant@example.com", resume: "resume.pdf", coverLetter: "cover letter text" }
 - Responses: 201, 400, 404
- *GET /applications/<id>*
 - Query Param: applicationID
 - Responses: 200, 404
- *GET /applications*
 - Query Param: applicantID
 - Responses: 200, 404

interaction/ Interaction and Tracking Service (protected)

Service Description: Handles interactions such as application status updates and tracking.

Endpoints:

- *GET /status/<applicationID>*
 - Query Param: applicationID
 - Responses: 200, 404
- *PUT /status/<applicationID>*
 - Payload: { status: "Reviewed" }
 - Responses: 204, 401, 404
- *GET /track/<jobID>*
 - Query Param: jobID
 - Responses: 200, 404
- *POST /notify*
 - Payload: { applicantID: "applicant@example.com", message: "Your application status has been updated." }
 - Responses: 201, 400, 404

admin/ Admin Service (protected)

Service Description: Handles administrative functions related to user and job management.

Endpoints:

- *GET /users*
 - Query Param: userID

- Responses: 200, 404
- *DELETE /users/<id>*
 - Query Param: userID
 - Responses: 204, 404
- *GET /postings*
 - Query Param: adminID
 - Responses: 200, 404
- *DELETE /postings/<id>*
 - Query Param: jobID
 - Responses: 204, 404

reporting/ Reporting Service (protected)

Service Description: Provides reporting and analytics functionalities for job postings and application statuses.

Endpoints:

- *GET /dashboard*
 - Query Param: userID
 - Responses: 200, 404
- *GET /reports*
 - Query Param: userID
 - Responses: 200, 404
- *POST /reports*
 - Payload: { reportType: "summary", period: "monthly" }
 - Responses: 201, 400, 404