BREAKWATERS RECRUITING

Keagan Boucher 241260

Table of Contents

BREAKWATERS RECRUITING....1
Keagan Boucher 241260....1
B) Client Conceptualization & Problem statement....4
Overview of the project and business domain....4
The core problem or need the app will solve....4
Why the software solution is necessary....5
Any constraints or limitations you foresee....5
Develop a name and a basic logo for your management system....6
C) System Architecture....7
High-level system design diagram....7
Technologies & frameworks....8
Frontend: React.js....8
Backend: Node.js with Express.js....8
Database: MySQL....9
Authentication and Security....9
File Handling and Storage....10
Email Notifications....10
Justification for your tech stack selection....11
React.js (Frontend)....11
Node.js + Express.js (Backend)....11
MySQL (Database)....11
JWT Authentication....11
Multer or Cloud Storage....11
D) Feature Requirements & Scope....12
Define the scope of your system....12
Define SMART objectives for your project....13
List of major features....13
Feature Prioritisation....14
Define user roles (e.g. Admin, Standard User, Read-Only User)....14
System Flow Diagrams / User Stories....14
Key Use Case 1: Client Submits CV....14
Key Use Case 2: Company Views Assigned Candidates....15
E) Data Planning....16
Entity-Relationship Diagram (ERD)....16
Explanation of key tables and their relationships....17
1. users....17
2. clients....18
3. companies....18
4. cvs....19
5. assignments....19
6. activity_logs....20
Data types and constraints for key fields....20

F) Wireframes & UI/UX Considerations.....21
Moodboard or visual inspiration for UI styling and brand identity.....21
Basic wireframes of key screens.....22
Home.....22
Login.....22
Dashboard.....23
CV Upload.....23
Justification of frontend frameworks, styling choices, and usability considerations.....24
Justification of Frontend Frameworks and Styling Choices.....24
Accessibility Concerns.....25
Usability Considerations.....25
Accessibility Considerations.....25
G) Project Timeline & Workflow.....26
High-level Gantt chart or roadmap of project phases.....26
Milestones & Deliverables Breakdown.....26
Estimated Timelines by System Layer.....27
Project Management Approach.....27
H) Risks, Challenges & Conclusion.....28
Potential Risks and Challenges.....28
Conclusion.....28

B) Client Conceptualization & Problem statement

Overview of the project and business domain

Breakwaters is a modern recruitment management system that acts as a bridge between aspiring job seekers and hiring companies. The platform is designed to simplify and centralize the recruitment process by allowing clients (job seekers) to submit their CVs and relevant personal information. A dedicated recruitment officer then reviews these submissions and strategically matches each candidate with a registered company based on their needs, culture, and role availability.

The project exists within the recruitment and human resources technology domain, serving as a niche tool for small to medium enterprises (SMEs) and candidates seeking tailored job placements rather than bulk-automated hiring processes. It blends human decision-making with digital efficiency to improve job placement success and workplace compatibility.

The core problem or need the app will solve

Traditional recruitment processes are often fragmented, impersonal, and inefficient. Job seekers may submit countless CVs into automated systems that don't truly evaluate their strengths or match them with the right roles. Likewise, companies often struggle to find high-quality candidates that not only meet job requirements but also align with their culture.

Breakwaters solves this by:

1. Giving candidates a centralized, guided platform to submit their profiles.
2. Empowering recruitment officers to act as human curators, matching candidates manually but efficiently to relevant companies.
3. Giving companies access to pre-vetted candidate pools rather than raw, unfiltered applications.

Why the software solution is necessary

The employment landscape is becoming increasingly saturated with both underqualified and overqualified candidates applying to the same roles. Generic job boards and automated recruitment systems fail to assess the nuance behind a CV or understand a candidate's potential and career goals.

This system is necessary because:

- It **humanizes** the recruitment process.
- It reduces the **burden on companies** by offloading vetting to recruitment officers.
- It offers a **personalized experience** for job seekers.
- It increases **placement success** through targeted matching.

Additionally, Breakwaters introduces a curated ecosystem of talent and opportunity. Focusing on quality over quantity.

Any constraints or limitations you foresee

Data Protection & Privacy: Handling CVs and personal information requires strict adherence to data protection laws such as **POPIA (South Africa)** or **GDPR** (if expanding internationally). Consent and secure storage will be a legal and technical priority.

Scalability of Manual Matching: Since matching is performed by a recruitment officer, scaling the platform to accommodate large volumes of users may become a bottleneck unless supported by admin tools or assistant automation in the future.

File Storage Limitations: CVs will be uploaded as documents, which could raise storage and server load concerns unless limited to a specific file size and type (e.g., PDF)

Verification of Companies: To ensure safety and legitimacy, new company registrations will need to be reviewed or approved manually by an admin before gaining access to the candidate database.

Develop a name and a basic logo for your management system

Project Name:

Breakwaters

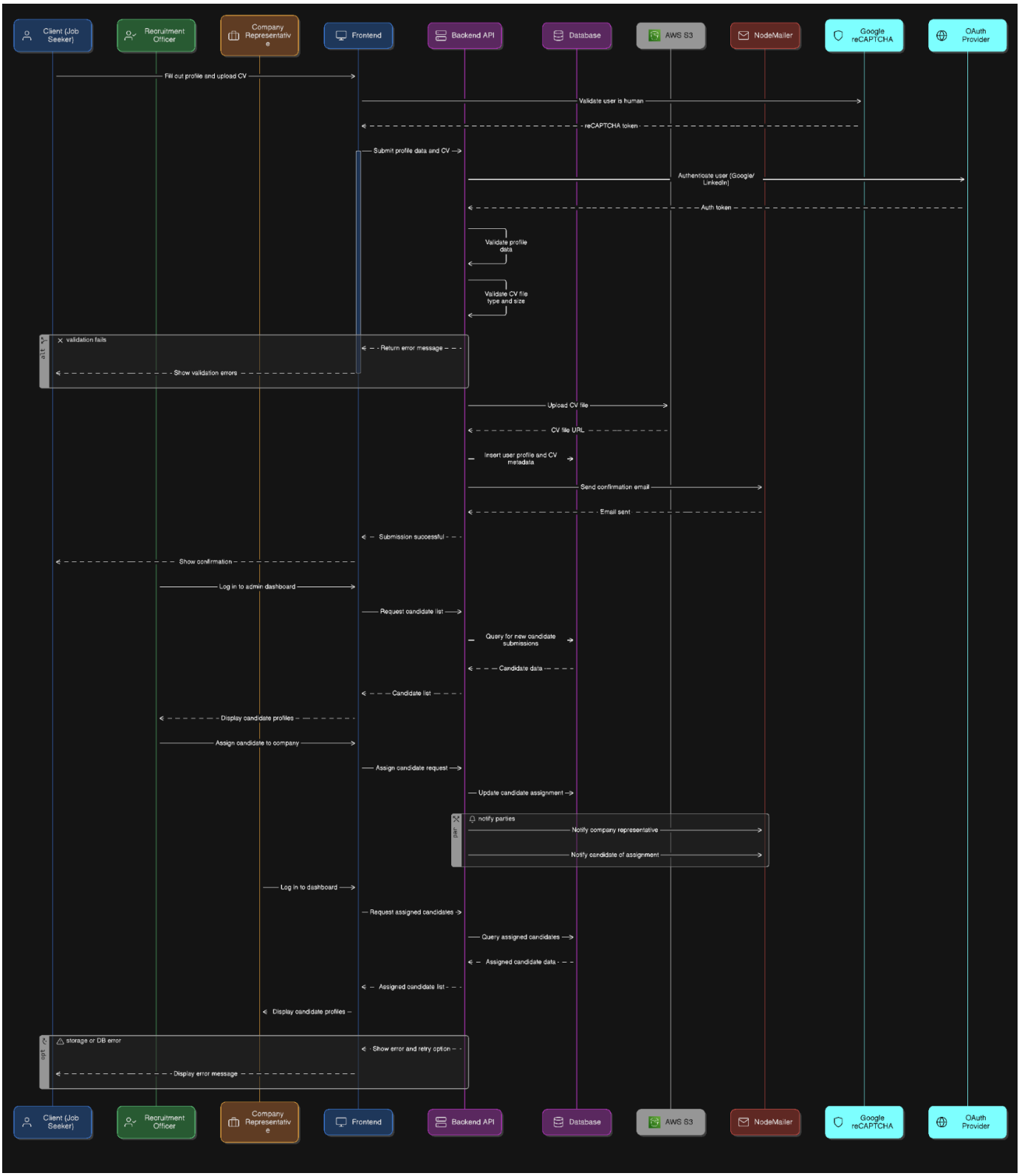
Name Rationale:

A breakwater is a structure that protects the shore from rough waves. Symbolically, *Breakwaters* protects job seekers from the chaotic and overwhelming tides of the job market, guiding them safely toward fitting employment opportunities.

Logo Concept:

C) System Architecture

High-level system design diagram



Technologies & frameworks

Frontend: React.js

Purpose:

To build a responsive and dynamic single-page application (SPA) for all user roles.

Justification:

- Component-based structure supports reusable and maintainable UI development
- Enables fast, interactive user interfaces with client-side routing
- Well-documented and widely supported in the developer community
- Integrates easily with styling libraries like Tailwind CSS or Bootstrap

Backend: Node.js with Express.js

Purpose:

To handle server-side logic, routing, API endpoints, user authentication, and middleware.

Justification:

- Non-blocking, event-driven architecture ideal for handling concurrent user requests
- Express simplifies API development with routing and middleware layers
- Works seamlessly with React through RESTful API communication

Database: MySQL

Purpose:

To manage structured data including users, roles, CV metadata, and assignment records.

Justification:

- Relational structure fits the system's needs for managing user relationships and data integrity
- Supports ACID transactions for reliable and consistent operations
- Offers flexibility for future scaling and hosting, whether self-managed or cloud-based (e.g., PlanetScale)

Authentication and Security

Technologies: JSON Web Tokens (JWT), Bcrypt

JWT:

- Implements stateless authentication across the application
- Encodes user role and ID, enabling role-based access to protected routes

Bcrypt:

- Ensures secure password storage by hashing user credentials before saving to the database

File Handling and Storage

Technologies: Multer (for file upload), AWS S3 or Cloudinary (for cloud storage)

Use Case:

- Client CVs are uploaded via Multer and optionally stored in a secure cloud storage service
- File metadata is stored in MySQL for retrieval and access control

Email Notifications

Technologies: NodeMailer (for development) or SendGrid (for production)

Use Case:

- Sends confirmation emails to clients after CV submission
- Notifies companies of candidate assignments
- Delivers password reset instructions when requested

Justification for your tech stack selection

React.js (Frontend)

- Enables rapid development using components for forms, dashboards, and profile views.
- Offers efficient client-side routing and reactivity.
- Supports scalable UI/UX as the platform grows (responsive + dynamic states).

Node.js + Express.js (Backend)

- Ideal for handling multiple API requests concurrently.
- Large ecosystem and middleware support for authentication, file uploads, and logging.
- RESTful architecture makes integration with React seamless.

MySQL (Database)

- A relational database is perfect for structured entities like users, companies, CVs, and placements.
- Mature and stable, with widespread support and hosting options.
- Ideal for enforcing constraints (foreign keys, unique values) and structured queries.

JWT Authentication

- Stateless and secure for handling multiple types of users (clients, companies, recruitment officers).
- Scalable for API-first systems where session management is critical.

Multer or Cloud Storage

- Multer handles file uploads securely on the backend.

D) Feature Requirements & Scope

Define the scope of your system

What is Included (In Scope for MVP):

- User registration and authentication (JWT-based)
- Role-based access for:
    - Clients (Job Seekers)
    - Recruitment Officers (Admins)
    - Company Representatives
- CV upload system (PDF only, with file size limitation)
- Recruitment officer dashboard to:
    - Review candidate submissions
    - Assign candidates to companies
- Company dashboard to view assigned candidates
- Email notifications:
    - CV upload confirmation
    - Candidate assignment alerts
    - Password reset links
- Secure MySQL database to store user data and candidate-company relationships

What is Excluded (Out of Scope for MVP):

- In-platform messaging or chat between clients and companies
- AI-driven candidate matching or recommendations
- Public job board or open search for all candidates
- Admin reporting dashboards or analytics
- Resume parsing or keyword analysis

Define SMART objectives for your project

Objective	Criteria
Implement secure JWT-based user authentication for all roles	Specific, Measurable
Allow registered clients to upload a CV and profile within 3 steps	Achievable, Relevant
Enable recruitment officers to assign at least one candidate to a company by week 6	Time-bound, Measurable
Develop role-based dashboards tailored to each user type	Specific, Achievable
Achieve MVP deployment within 10 weeks using GitHub and Trello for management	Time-bound, Realistic

List of major features

Feature Category	Feature Description
Authentication	User login, registration, and secure token-based access control
User Roles	Role-specific permissions for clients, recruitment officers, companies
CV Management	CV upload (PDF), file storage, metadata tagging
Candidate Review	Admin dashboard to browse, filter, and assign candidates
Company Access	Dashboard to view assigned candidates, mark interest, submit feedback
Email Service	Confirmation and notification emails triggered on key actions
Data Storage	Structured relational database (MySQL) to handle all user and CV data

Feature Prioritisation

Priority	Features
MVP (Must Have)	User authentication, CV upload, role-based dashboards, candidate assignment, basic email notifications
Nice-to-Have	Company feedback on candidates, admin filtering by skill/location, user profile images
Future Considerations	AI-driven matching, resume parsing, in-app messaging, job board listing, analytics dashboard, external OAuth login

Define user roles (e.g. Admin, Standard User, Read-Only User)

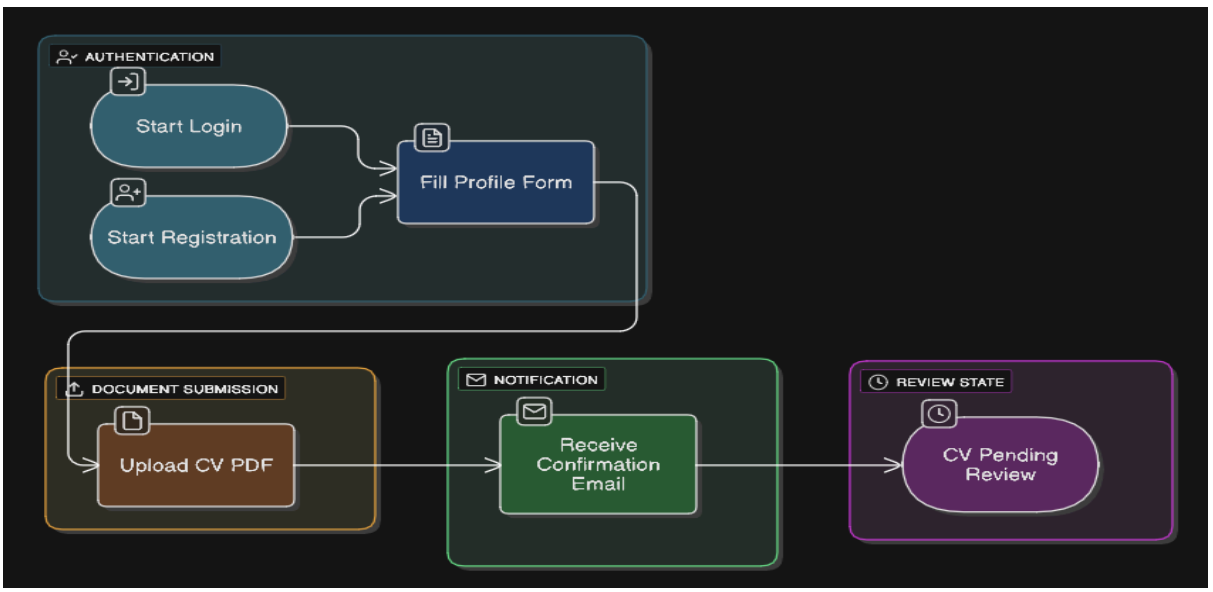
Role	Capabilities
Client (Job Seeker)	Register, log in, submit CV, update profile, delete account
Recruitment Officer (Admin)	Access all candidates and companies, assign matches, moderate activity
Company Representative	View only assigned candidates, mark interest, provide feedback to admin

System Flow Diagrams / User Stories

Key Use Case 1: Client Submits CV

- **User Story:**

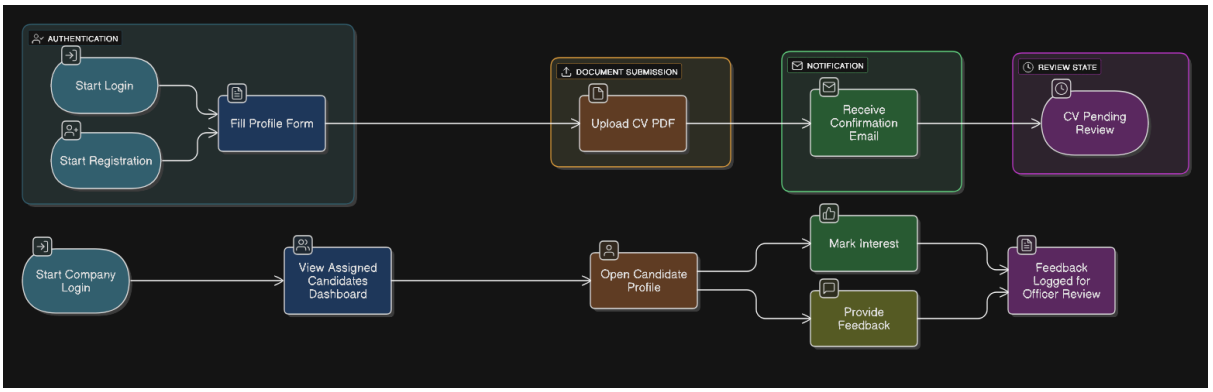
  As a job seeker, I want to upload my CV and describe my skills so I can be matched with a company.



Key Use Case 2: Company Views Assigned Candidates

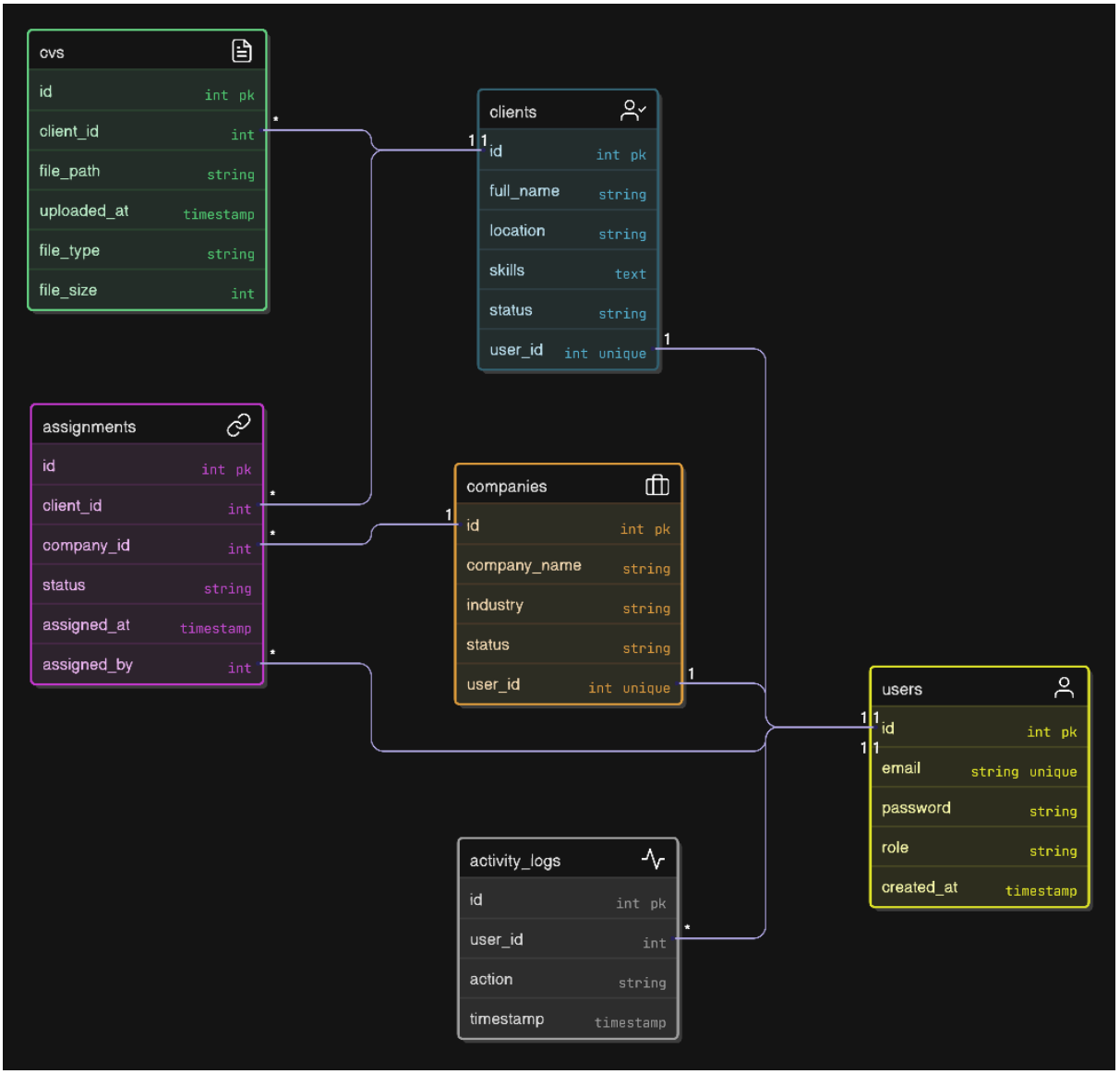
- **User Story:**

  As a company representative, I want to view candidates assigned to me so I can evaluate and shortlist them.



E) Data Planning

Entity-Relationship Diagram (ERD)



Explanation of key tables and their relationships

1. users

- Stores login credentials and role information for all users
- Roles: `"client"`, `"company_rep"`, `"recruitment_officer"`

Field	Type	Constraints
id	INT (PK)	AUTO_INCREMENT
email	VARCHAR(255)	UNIQUE, NOT NULL
password	VARCHAR(255)	Hashed with Bcrypt, NOT NULL
role	ENUM	('client', 'company_rep', 'recruitment_officer')
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

Relationships:

- One-to-one with `clients` (if role is "client")
- One-to-one with `companies` (if role is "company_rep")

2. clients

- Stores extended profile data for job seekers

Field	Type	Constraints
id	INT (PK)	AUTO_INCREMENT
user_id	INT (FK)	REFERENCES users(id), UNIQUE
full_name	VARCHAR(255)	NOT NULL
location	VARCHAR(100)	NULLABLE
skills	TEXT	NULLABLE (comma-separated)
status	ENUM	('pending', 'assigned', 'rejected') DEFAULT 'pending'

Relationships:

- One-to-one with `users`
- One-to-many with `cvs`
- One-to-many with `assignments`

3. companies

- Stores information about registered companies

Field	Type	Constraints
id	INT (PK)	AUTO_INCREMENT
user_id	INT (FK)	REFERENCES users(id), UNIQUE
company_name	VARCHAR(255)	NOT NULL
industry	VARCHAR(100)	NULLABLE
status	ENUM	('unverified', 'verified') DEFAULT 'unverified'

Relationships:

- One-to-one with `users`
- One-to-many with `assignments`

4. cvs

- Stores uploaded CV metadata

Field	Type	Constraints
id	INT (PK)	AUTO_INCREMENT
client_id	INT (FK)	REFERENCES clients(id)
file_path	VARCHAR(255)	NOT NULL (points to S3/local)
uploaded_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
file_type	VARCHAR(50)	CHECK ('.pdf') only
file_size	INT	Max 5MB (~5,242,880 bytes)

5. assignments

- Connects candidates to companies and tracks assignment status

Field	Type	Constraints
id	INT (PK)	AUTO_INCREMENT
client_id	INT (FK)	REFERENCES clients(id)
company_id	INT (FK)	REFERENCES companies(id)
assigned_by	INT (FK)	REFERENCES users(id) (recruitment officer)
status	ENUM	('assigned', 'interview', 'declined', 'hired')
assigned_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

6. activity_logs

- Logs actions like uploads, logins, assignments, etc.

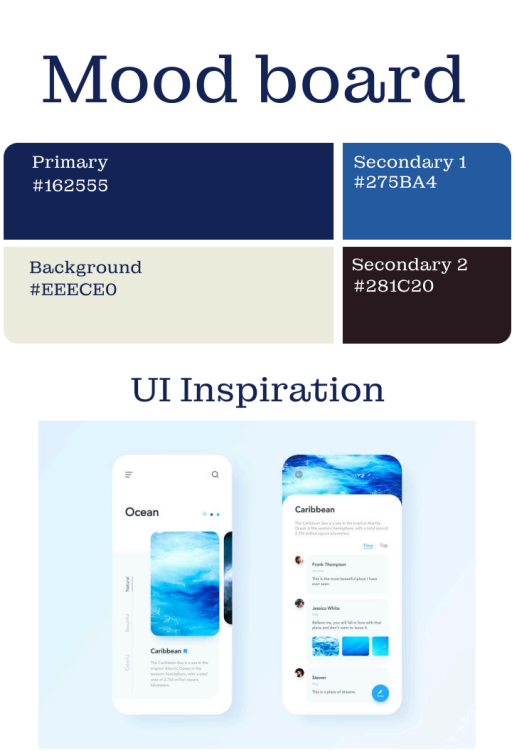
Field	Type	Constraints
id	INT (PK)	AUTO_INCREMENT
user_id	INT (FK)	REFERENCES users(id)
action	VARCHAR(255)	NOT NULL
timestamp	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

Data types and constraints for key fields

Field Type	Description
INT	Used for primary keys and foreign keys
VARCHAR(n)	Used for text fields with limited length (email, name)
TEXT	Used for larger text blocks (skills, feedback)
ENUM	Used to enforce limited option values (roles, statuses)
TIMESTAMP	Used for tracking created/updated times
BOOLEAN	Optional for flags like "verified", "active", etc.

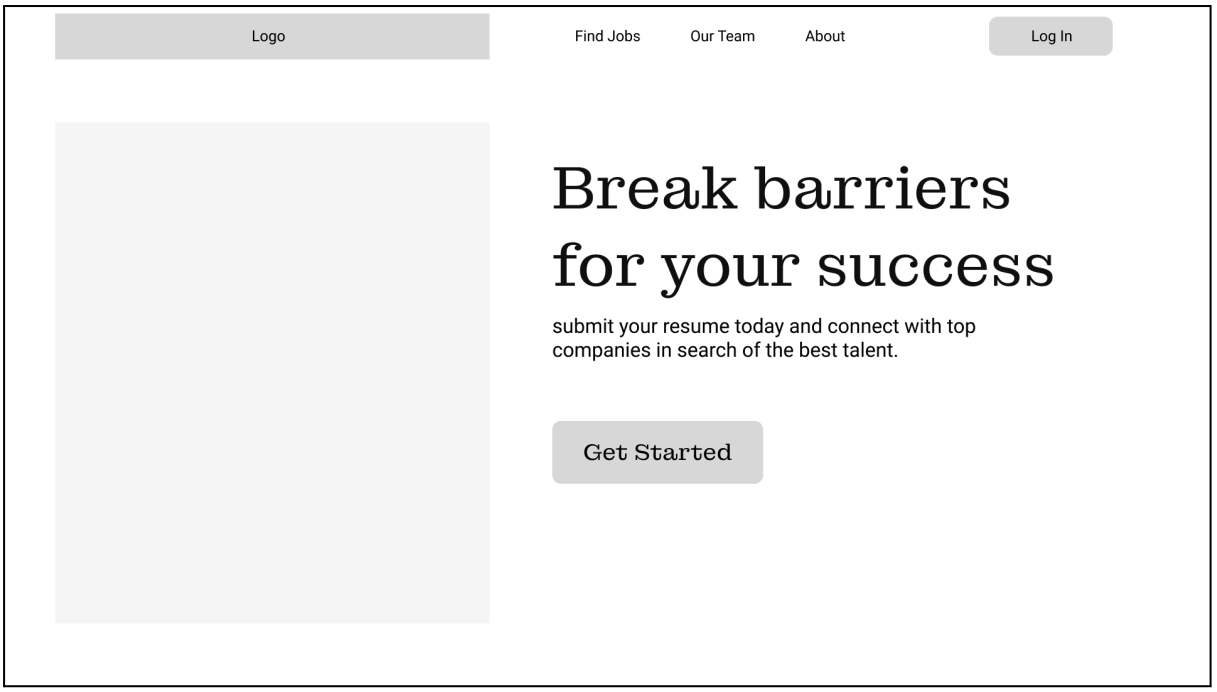
F) Wireframes & UI/UX Considerations

Moodboard or visual inspiration for UI styling and brand identity

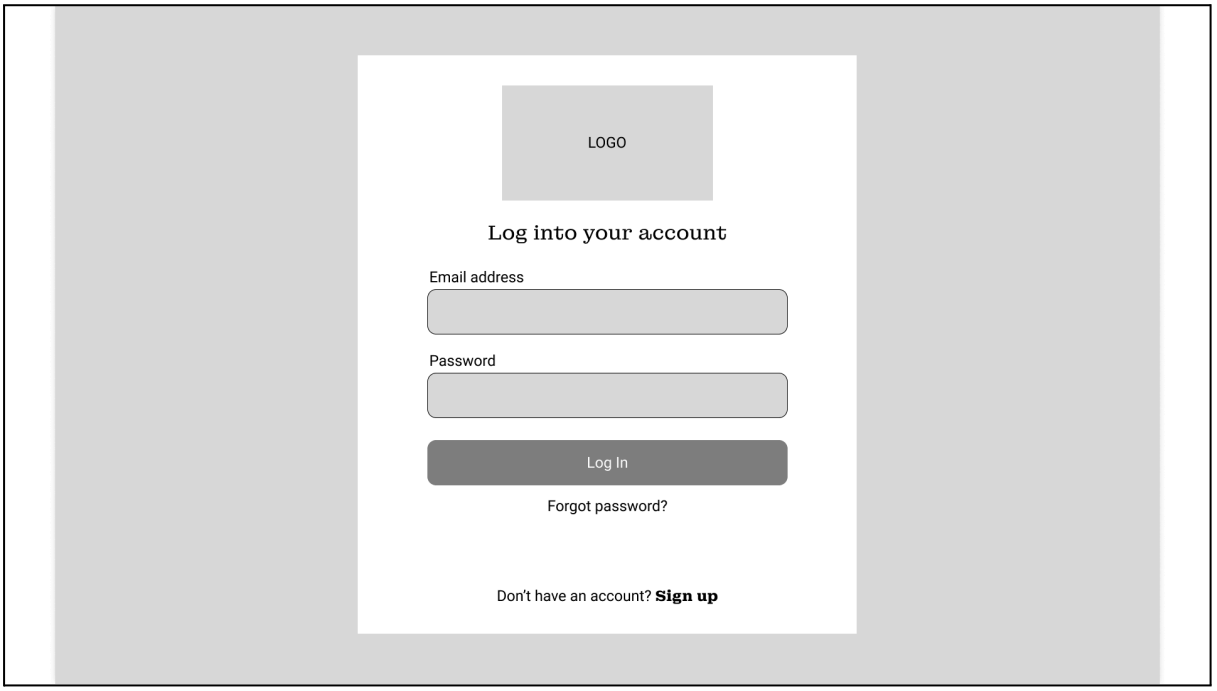


Basic wireframes of key screens

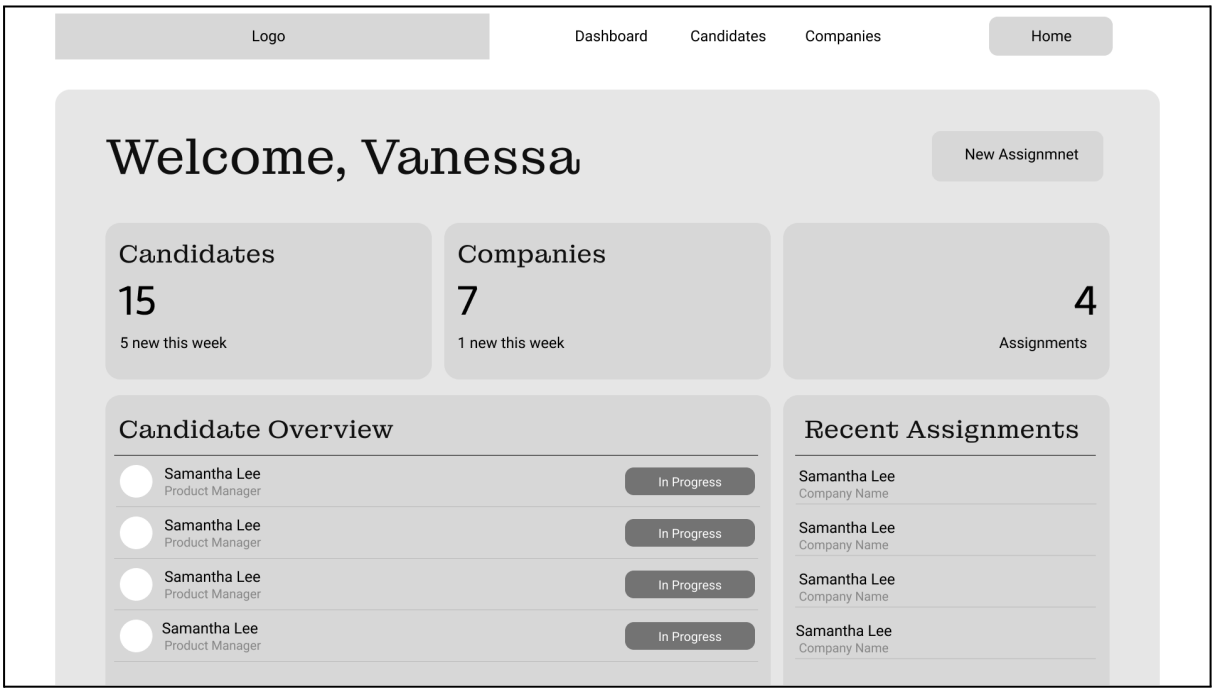
Home



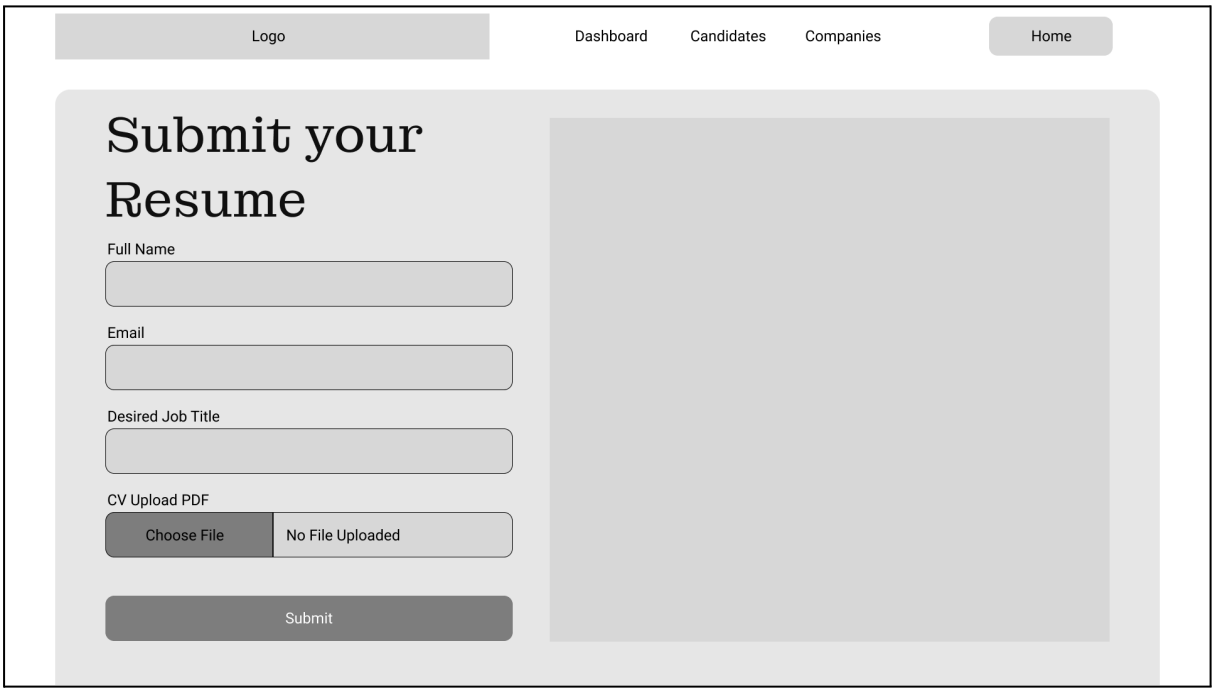
Login



Dashboard



CV Upload



Justification of frontend frameworks, styling choices, and usability considerations

Justification of Frontend Frameworks and Styling Choices

React.js was selected for the frontend due to its component-based architecture, which is ideal for reusable UI elements such as cards, forms, and dashboards. It also enables dynamic state updates, which support real-time dashboard metrics and conditional rendering based on user roles (client, recruitment officer, company rep).

Styling Choices:

- **Color Scheme**: Informed by the mood board, the site uses a calm and professional palette (navy, ocean blues, off-white) to reflect trust, clarity, and approachability.
- **Typography**: A strong, elegant serif typeface is used for headings to convey credibility and tradition, while a clean sans-serif font is used for body content for readability.
- **Layout**: Uses whitespace effectively to separate sections and prevent cognitive overload. Grids and card-based UI modules make the design scalable across screen sizes.
- **Visual Elements**: Ocean-inspired imagery and minimalist line illustrations tie the brand's identity to its metaphor — breaking through career barriers like waves on breakwaters.

Accessibility Concerns

Usability Considerations

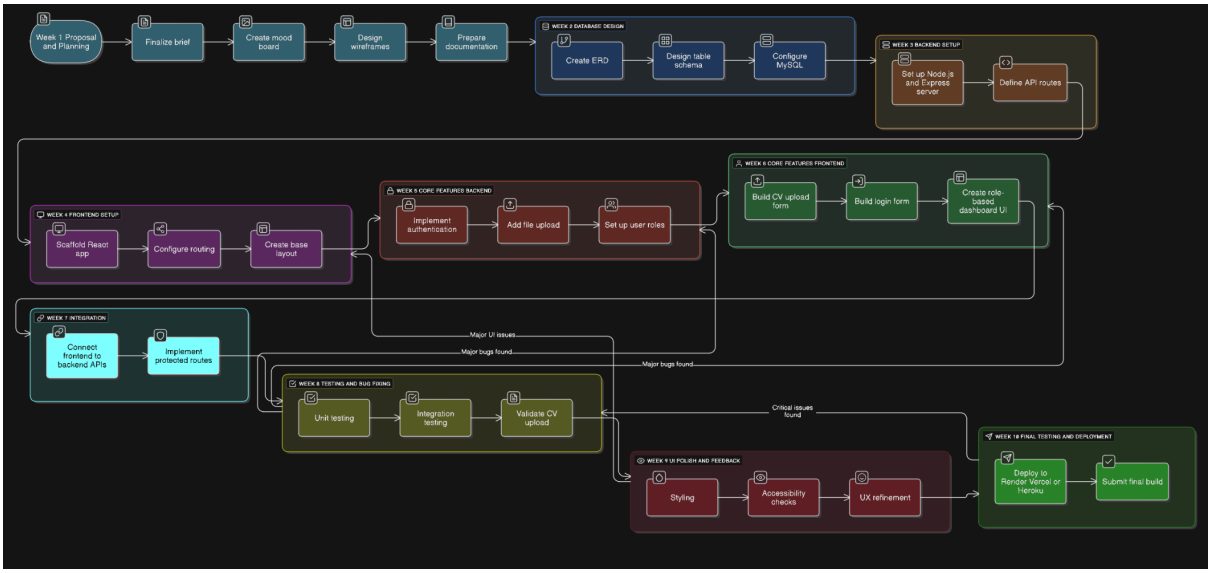
- **Clear Call-to-Action**: Every page includes an intentional next step for the user (e.g., “Submit CV,” “Log In,” “New Assignment”).
- **Dashboard Clarity**: Key metrics are surfaced at the top of the dashboard with bold typography, reducing the need for excessive clicks.
- **Consistent Navigation**: A persistent top nav bar allows users to access primary areas (Dashboard, Candidates, Companies) without friction.

Accessibility Considerations

- **Color Contrast**: All typography and interface elements will meet or exceed WCAG AA contrast ratios for readability.
- **Keyboard Navigation**: Interactive elements such as buttons, links, and input fields will be fully keyboard-navigable.
- **Form Labels and ARIA**: All input fields will include clear labels, and ARIA attributes will be used where appropriate to assist screen readers.
- **Responsive Layout**: UI will adapt gracefully to mobile, tablet, and desktop formats using responsive grid and flexbox layouts.

G) Project Timeline & Workflow

High-level Gantt chart or roadmap of project phases



Milestones & Deliverables Breakdown

Milestone	Week Due	Deliverable
System Proposal Approval	Week 1	PDF proposal and ERD
Database and API Structure Finalized	Week 3	Fully designed schema with working endpoints
Auth + CV Upload Functionality	Week 5	Working user roles, secure login, CV upload
Role-Based Dashboards	Week 6	Candidate and Admin dashboards complete
Frontend + Backend Integrated	Week 7	Live data exchange between UI and server
Testing Suite Completed	Week 8	All main flows tested and validated
Final Deployment	Week 10	Live system, submission-ready

Estimated Timelines by System Layer

System Area	Estimated Timeframe
Backend	Weeks 3–5
Frontend	Weeks 4–6
Integration	Week 7
Testing	Weeks 8–9
Deployment	Week 10

Project Management Approach

Methodology:

Agile-inspired workflow with weekly sprints and midweek reviews. Development tasks will follow a **Kanban board structure** to prioritize clarity and flexibility.

Tools Used:

- **GitHub** – Version control and issue tracking
- **Notion**– Task tracking and sprint organization
- **Figma** – Wireframes and prototype iteration
- **VS Code** – Development environment
- **Postman**– API testing

H) Risks, Challenges & Conclusion

Potential Risks and Challenges

Type	Risk	Mitigation Strategy
Technical	Data security breaches or unauthorized access	Implement strong authentication with JWT, enforce role-based access control (RBAC), hash passwords using Bcrypt, and validate all file uploads.
Technical	Scalability of file storage for CV uploads	Limit file size and type (e.g., 5MB PDFs), and integrate with scalable cloud storage like AWS S3 after MVP.
Technical	Backend/API integration failures	Use Postman for early endpoint testing, maintain modular code, and implement thorough integration testing in week 8.
Non-Technical	Delays in development timeline due to workload or scope creep	Use Notion to manage tasks weekly, limit MVP to essential features only, and hold regular sprint reviews.
Non-Technical	Difficulty verifying real companies or preventing misuse	Require company registration to be reviewed and verified by a recruitment officer before access is granted.
Non-Technical	Users not understanding how to navigate the platform	Create a minimalist UI with clear calls to action and ensure usability through wireframe testing and feedback.

Conclusion

Breakwaters offers a targeted, human-centered approach to solving the inefficiencies of the current job application process—especially for small and medium-sized enterprises. Unlike generic job boards or fully automated recruitment tools, Breakwaters introduces a balanced model that combines digital infrastructure with manual curation by recruitment officers. This ensures that candidates are not only technically qualified but culturally aligned with the companies they are matched to. Breakwaters provides a seamless and professional experience for all users. With a strong technical foundation, clear scope, and a focused MVP, the system is well-positioned to evolve into a practical, scalable solution to modern recruitment challenges.