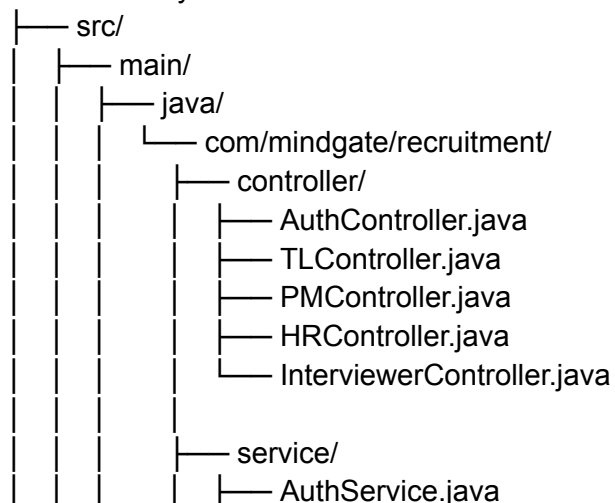
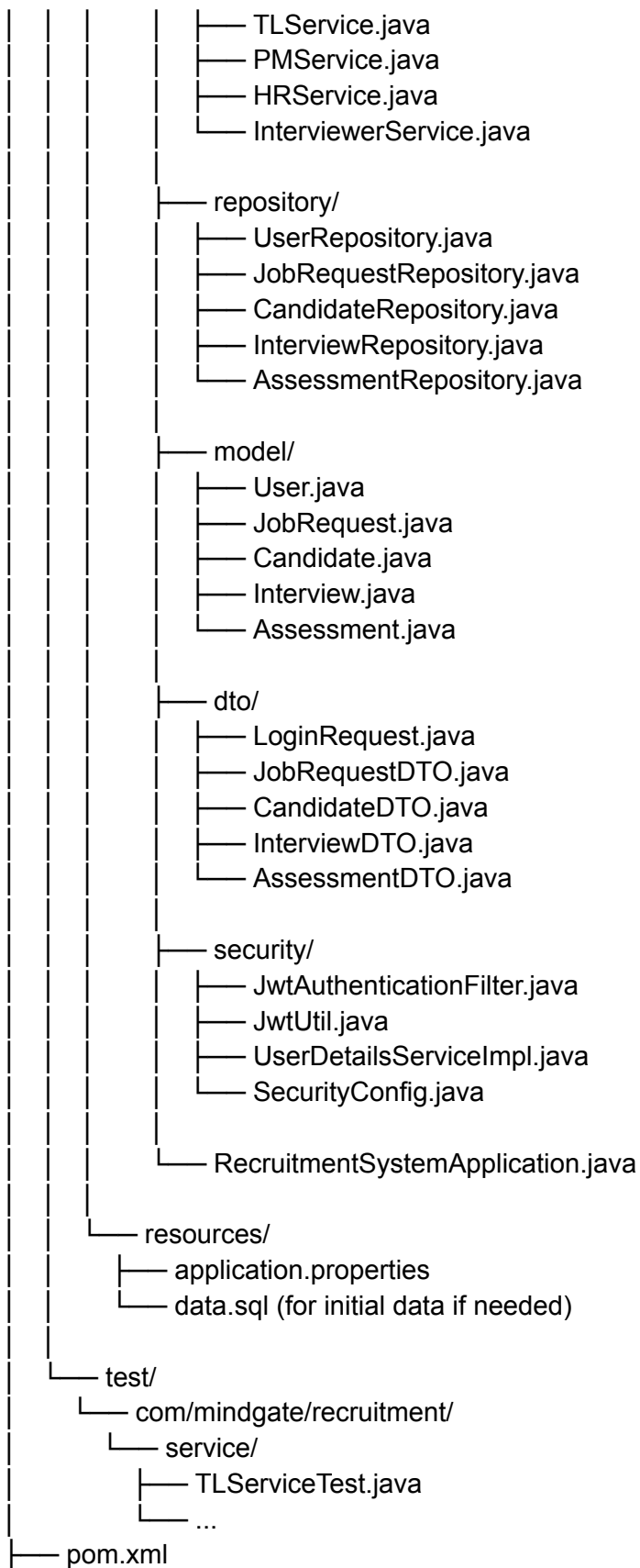


Role	Responsibilities
Team Leader (TL)	<ul style="list-style-type: none"> - Login - Create/edit job descriptions (JD) - Submit recruitment requests - Track status of submitted requests
Project Manager (PM)	<ul style="list-style-type: none"> - Login - View job requests - Analyze budget - Check internal resource availability - Assign project if resource available or forward request to HR
HR	<ul style="list-style-type: none"> - Post JDs - Manage applicant list - Filter candidates - Add new applicants - Conduct salary negotiation - Send emails (invitation, confirmation, docs) - Close or forward requests
Interviewer	<ul style="list-style-type: none"> - Login - Conduct technical interview - Submit assessment (selected/not selected)

✓ 1. Spring Boot Backend Structure (Java)

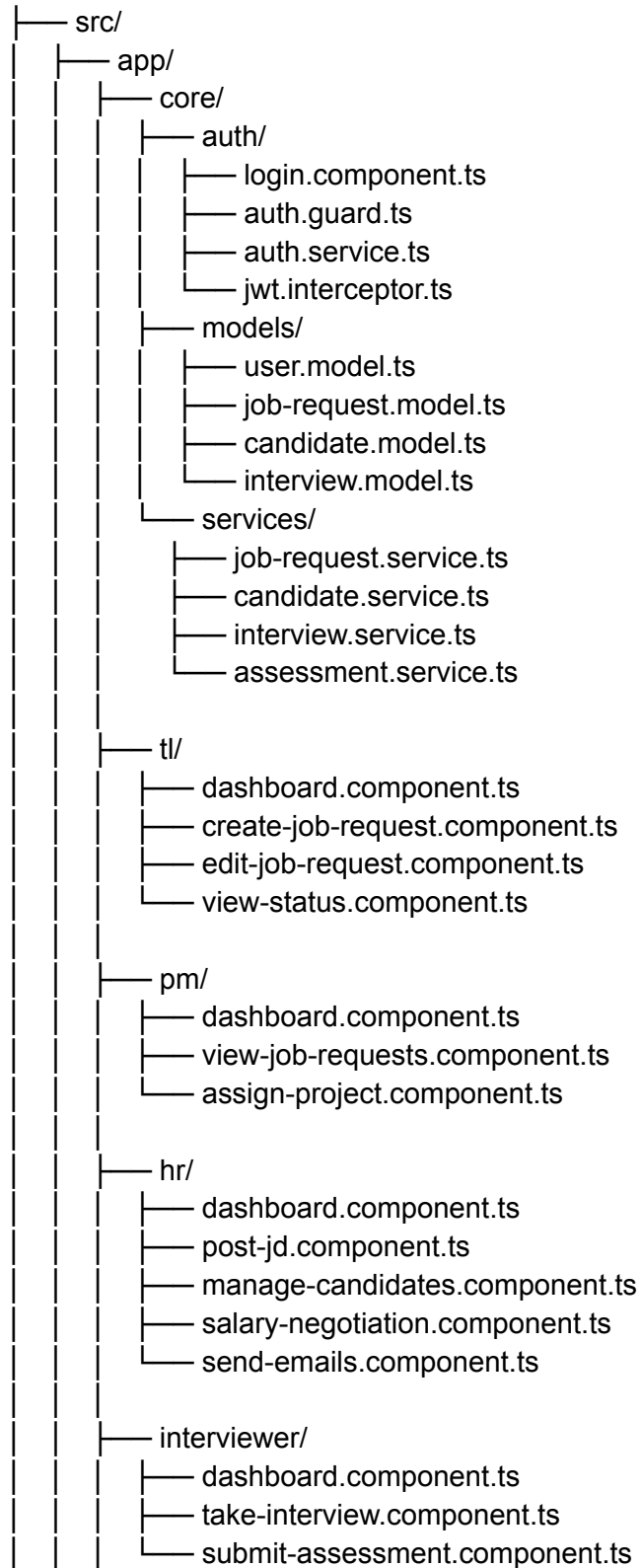
recruitment-system-backend/

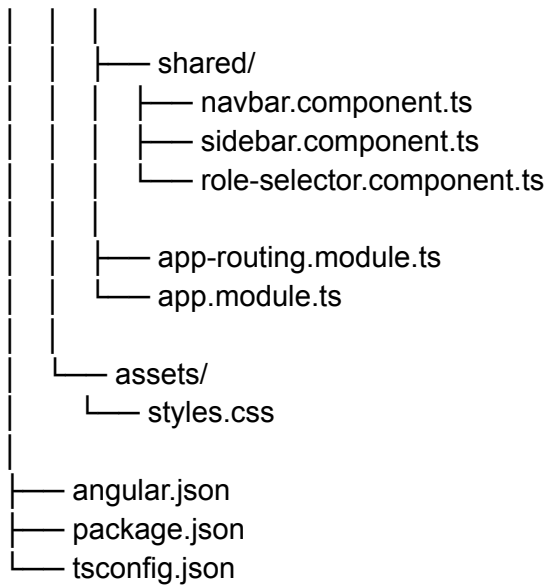




✓ 2. Angular Frontend Structure

recruitment-system-frontend/





1. EmployeeTable

Purpose: Stores login credentials and role for each system Employee (TL, PM, HR, INTERVIEWER).

Field	Type	Description
id	INT	Primary key, auto-increment
username	VARCHAR	Unique username for login
password	VARCHAR	Hashed password
email	VARCHAR	Optional email
role	ENUM	Defines the user's role
created_at	TIMESTAMP	Audit field

 **Why:** Every person accessing the system must have a unique account with an associated role.

2. **job_requests** Table

Purpose: Created by TLs to request new candidates for a project.

Field	Type	Description
<code>id</code>	<code>INT</code>	Primary key
<code>title</code>	<code>VARCHAR</code>	Job title
<code>description</code>	<code>TEXT</code>	Full job description
<code>required_experience</code>	<code>INT</code>	Experience needed
<code>required_skills</code>	<code>TEXT</code>	Skills required
<code>status</code>	<code>ENUM</code>	Tracks lifecycle: PENDING, ASSIGNED, etc.
<code>created_by</code>	<code>INT</code>	Foreign key to <code>users(id)</code>
<code>created_at</code>	<code>TIMESTAMP</code>	Submission date

 **Why:** TLs define project needs. PMs or HRs then take action.

3. **candidates** Table

Purpose: Stores information about job applicants.(HR)

Field	Type	Description
id	INT	Candidate ID
full_name, email, phone	VARCHAR	Basic info
experience, skills	TEXT/INT	Candidate profile
status	ENUM	NEW, SELECTED, REJECTED, etc.
job_request_id	INT	Links to the job request
added_by	INT	HR who added candidate
created_at	TIMESTAMP	Timestamp

✅ **Why: Central record of all applicants to track their journey.**

4. interviews Table

Purpose: Stores scheduled interviews for candidates.

Field	Type	Description
id	INT	Interview ID
candidate_id	INT	Who is being interviewed

interviewer_id	INT	Who is conducting interview
interview_date	DATETIME	Scheduled date
comments	TEXT	Interviewer's remarks
result	ENUM	SELECTED, NOT_SELECTED, PENDING
created_at	TIMESTAMP	Timestamp

✅ **Why: Tracks interview details and final result.**

5. **assessments** Table

Purpose: Stores technical scores and evaluation remarks by interviewer.

Field	Type	Description
id	INT	Primary key
interview_id	INT	Links to the interview
score	INT	Interview rating (0–100)
remarks	TEXT	Final decision notes
created_at	TIMESTAMP	Timestamp

✅ **Why: Needed for performance review and transparency in selection.**

6. **salary_negotiations** Table (Optional, useful for HR)

Purpose: Manages salary discussion results between HR and selected candidate.

Field	Type	Description
candidate_id	INT	Candidate being negotiated with
expected_salary, offered_salary	DECIMAL	Values in INR or USD
status	ENUM	AGREED, DECLINED, etc.
negotiated_by	INT	HR User ID
updated_at	TIMESTAMP	Timestamp

✓ **Why:** Tracks salary discussions for compliance and history.

7. **email_logs** Table (Optional)

Purpose: Logs emails sent (invitations, confirmations, etc.)

Field	Type	Description
to_email, subject, message	VARCHAR/TEXT	Email content
sent_by	INT	User ID (usually HR)
sent_at	TIMESTAMP	Timestamp

✓ **Why:** Good for audit trail and debugging email issues.

How It All Connects

- `users` table manages access control.
- `TL` users submit `job_requests`.
- `PM` users handle the next step: assign from bench or forward to `HR`.
- `HR` users manage `candidates`, post JDs, and send interview invites.
- `INTERVIEWER` users handle `interviews` and submit `assessments`.
- All actions are timestamped and trackable.