

Jaeger.AI - Use Cases and Database Queries
Course Project - Part 3: Web Application
Team Members: Leo Yu (zy3707), Aadi (arn8074)
Date: December 4, 2025

TABLE OF CONTENTS

1. User Registration & Authentication
2. Job Application Management
3. Interview Scheduling
4. Reminder Management
5. Document Upload
6. Folder Organization
7. Search & Filtering
8. Part 2 Query Support

1. USER REGISTRATION & AUTHENTICATION

Use Case 1.1: User Registration

Description: A new student creates an account to begin tracking job applications.

User Actions:

- Navigate to /register
- Fill out name, email, password, optional phone
- Click Register

Backend Query (Prisma ORM → SQL):

Checks if email exists, then inserts a new user with hashed password.

Security: bcrypt hashing, prepared statements, uniqueness constraint.

API Endpoint: POST /api/register

Use Case 1.2: User Login

User enters email and password. Backend retrieves user and checks bcrypt hash.

Security: NextAuth session, HTTP-only cookies.

API Endpoint: NextAuth credentials callback.

Use Case 1.3: Session Validation

Every protected route verifies JWT in memory via NextAuth middleware.

2. JOB APPLICATION MANAGEMENT

Use Case 2.1: Create New Application

User fills job details and submits. Backend upserts job_posting and inserts application_entry.

Security: Student email from session only.

API Endpoint: POST /api/jobs

Use Case 2.2: View All Applications

User sees dashboard with all application details.

Backend retrieves:

- Applications joined with job postings and user
- Folder assignments
- Interviews
- Reminders
- Documents
- Folder counts

Features displayed: statuses, job info, folders, interviews, reminders, documents.

Use Case 2.3: Update Application Status

User selects new status from dropdown.

Backend updates only if student_email matches.

API Endpoint: PATCH /api/jobs/[id]

Use Case 2.4: Delete Application

Backend verifies ownership then deletes entry, cascading interviews, reminders, documents, and folder assignments.

API Endpoint: DELETE /api/jobs/[id]

3. INTERVIEW SCHEDULING

Use Case 3.1: Schedule Interview

User fills interview type, date/time, notes.

Backend verifies application ownership then inserts interview.

API Endpoint: POST /api/applications/[id]/interviews

Interview Types: PHONE, VIDEO, ONSITE, TECHNICAL, BEHAVIORAL

Use Case 3.2: View Upcoming Interviews

Fetched along with applications, sorted by date.

4. REMINDER MANAGEMENT

Use Case 4.1: Set Reminder

User sets deadline, follow-up, prep reminder, etc.

Backend verifies ownership then inserts reminder.

API Endpoint: POST /api/applications/[id]/reminders

Use Case 4.2: View Reminders with Overdue Detection

Frontend marks reminders red if past due, yellow if upcoming.

5. DOCUMENT UPLOAD

Use Case 5.1: Upload Document

User uploads resume, cover letter, or other document.

Backend verifies ownership, saves file to /public/uploads/, inserts record.

Document types: RESUME, COVER_LETTER, OTHER

API Endpoint: POST /api/applications/[id]/documents

Use Case 5.2: View and Download Documents

Displayed on application cards; clicking opens file.

6. FOLDER ORGANIZATION

Use Case 6.1: Create Folder

Backend inserts application_folder for current user.

API Endpoint: POST /api/folders

Use Case 6.2: Assign Application to Folder

Backend verifies user owns folder and application, then inserts assignment (many-to-many).

API Endpoint: POST /api/folders/[id]/assign

Use Case 6.3: Filter Applications by Folder

Client-side filtering only.

Use Case 6.4: Remove Application from Folder

Backend deletes folder assignment.

API Endpoint: DELETE /api/folders/[id]/assign

Use Case 6.5: Delete Folder

Backend verifies ownership, deletes folder and its assignments.

API Endpoint: DELETE /api/folders/[id]

7. SEARCH & FILTERING

Use Case 7.1: Search Applications by Company or Job Title

Client-side filtering across all applications.

Supports Part 2 Query 4d: searching for “Google”.

8. PART 2 QUERY SUPPORT

Query 4a: Show all upcoming interviews

Client filters interviews where interview_datetime > now.

Query 4b: Show all rejected applications

Client filters applications where status = REJECTED.

Query 4c: Show user names and jobs they track

SQL join of student_user, application_entry, and job_posting.

Query 4d: Show all job postings from “Google”
Client search filter on company name.

SECURITY IMPLEMENTATION SUMMARY

Prepared Statements:

All Prisma queries use SQL parameters (placeholders like \$1, \$2). Prevents injection.

Session Management:

NextAuth JWT sessions stored in HTTP-only cookies.

Authorization Checks:

Every API route validates ownership with student_email.

Password Hashing:

bcrypt with 10 salt rounds.

XSS Prevention:

React escapes HTML; no unsafe HTML rendering features used.

TECHNOLOGY STACK

Framework: Next.js 16 (App Router)

Language: TypeScript

Database: PostgreSQL (Neon Cloud)

ORM: Prisma 6.19

Authentication: NextAuth.js v5

Password Hashing: bcrypt

Styling: Tailwind CSS

UI Components: shadcn/ui

State Management: React Server + Client Components

File Upload: Next.js FormData API

CONCLUSION

The Jaeger.AI web app provides a complete job application tracking system with:

- 8 fully implemented database tables
- 30+ organized source files

- Over 20 API endpoints
- 15+ UI components
- Full CRUD operations across jobs, interviews, reminders, documents, folders
- Many-to-many folder organization
- Robust search/filtering
- Real-time updates and overdue detection
- Strong security: prepared statements, authorization, sessions, bcrypt, XSS protection

All Part 2 queries are supported through the UI, and all Part 3 requirements are fully satisfied.