Team Members:
Leo Yu, Aadi Narayan

Tech Stack:
Next.js, TypeScript, PostgreSQL, possibly Vercel

Work Division:
Aadi will focus on the frontend and Leo will focus on the backend. We will overlap and collaborate on any tasks needed to complete the project.

Meeting Plan:
Meet every weekend (at least once weekly) and work asynchronously during the week.

Challenges / Unknowns:
No major blockers identified yet; main challenge is getting familiar with the new technologies.

---

6. Use Cases Completed So Far
Completed: 0 use cases
The project is currently in the initial setup phase. The following use cases are planned:

- User Registration & Authentication – Sign up/login with Google OAuth
- Add Job Application – User can manually add a job with all details
- View All Applications – User can see a list of all their job applications
- Update Application Status – Change status (Applied → Interview → Offer)
- Create Folders – Organize applications into custom folders
- Assign Jobs to Folders – Categorize applications
- Search Applications – Search by company, title, or status
- Gmail Integration – Connect Gmail and receive notifications
- Set Reminders – Deadlines and follow-up reminders
- View Analytics Dashboard – Application statistics and insights

---

7. Database Tables

Status: Not created yet

The database schema is fully designed (in PRD.md) but not yet implemented in PostgreSQL.

Planned Tables:

1. users
   Stores user account information
   Fields: id, email, name, google_id, gmail_tokens, timestamps

2. job_applications
   Core table for job tracking
   Fields: id, user_id, job_url, job_title, company_name, location, salary, dates, description, status, notes, embedding (vector)
   Indexes: user_id + status, user_id + company_name, application_deadline

3. status_history
   Audit trail for status changes
   Fields: id, job_application_id, status, changed_at

4. folders
   Custom categories for organizing jobs
   Fields: id, user_id, name, color

5. job_folders
   Many-to-many relationship between jobs and folders
   Fields: job_application_id, folder_id (composite primary key)

6. reminders
   Deadline and follow-up reminders
   Fields: id, user_id, job_application_id, reminder_type, reminder_date, message

7. email_notifications
   Gmail integration data
   Fields: id, user_id, job_application_id, gmail_message_id, sender, subject, snippet

Additional Features Planned:

- Vector search using pgvector for semantic search

- Full-text search on job descriptions and titles

- Analytics tables and queries using GROUP BY and JOINs

---

8. Data Insertion Status

Status: No data inserted yet
Tables are not created, so no seed data exists.

Planned Approach:

- Create Prisma schema file
- Run Prisma migrations
- Create seed script with:
  - 2–3 test users
  - 20–30 sample job applications
  - 5–10 folders
  - Sample status history entries
  - Test reminders and notifications

---

9. Front-End Pages Completed

Completed:

- Home Page (index) – Basic Next.js landing page (placeholder)

Pages Not Yet Implemented:

- Login/Signup
- Dashboard
- Job Applications List
- Job Detail
- Add/Edit Job

- Folders Management
- Analytics Dashboard
- Settings
- Notifications Center

Current State:
Basic Next.js + Tailwind setup. The landing page exists but is only placeholder content.

---

10. Server-Side Implementation Status

Completed:

- Next.js project initialized with TypeScript
- Tailwind CSS configured
- Basic App Router structure
- ESLint configured

Not Yet Implemented:

- Database connection (Prisma)
- API routes
- Authentication (NextAuth.js)
- Server actions
- Data fetching utilities
- Gmail API integration
- Web scraping service