Description:

- Job Tracker is an innovative solution built to help people track their job applications efficiently and have real time analytics about their job hunting journey.

Functionalities:

Job applications:

- The user can save their job applications to keep track of them.
- The user can specify for each job application the company, the position, the location, the salary, the job description (or job description link) and the status (where the user is in the job application process)(interview, applied, saved only, offer, rejected).
- The user can filter through their applications by status.
- The user can look for their application by position or company name.

Interviews:

- The user can save their scheduled interviews.
- The user can specify for each interview the company, the position, the date, the time, the interview type (phone, on-site, online) and some details about it.
- The user can access to all their scheduled interview and filter them by date or type.

Profile:

- The user can create their own profile.
- The user can specify their full name, email, phone number, location, position and skills.
- The user can upload their resume.
- The user can add links to their social media.
- The user's profile is accessible by all the other users of the app.
- The user can look for other users using their name.
- The user can have a list of profiles they are interested in.

Contacts:

- The use can save multiple contacts with their information.
- The user can specify for each one the name, the position, the company they are working at and the contact information (phone number, email, social media).

- The contacts are not necessarily users of the application but if one of them is, a link is automatically added to access their profile.
- The user can search in their contacts by name, company or position.

Analytics:

- The user can have metrics concerning their applications.
- The user can see the number of applications submitted per month.
- The user can see the status of all the applications (how many are applied, how many are rejected, etc.).
- The user can see the distribution of applications in terms of positions (frontend, DevOps, etc.).
- The user can see the distribution of applications in terms of location.

Dashboard:

- The user can access a dashboard with an overview of their job applications.
- The user can see the total number of applications that are in progress.
- The user can see the upcoming interviews.
- The user can see the recent applications.

Classes:

- User
- Interviews
- Applications
- Contacts
- Analytics
- Notifications

Architecture:

For this project, we will implement a microservices' architecture.

Services:

Service	Responsibilities
User Service	Manages user accounts, profiles.
Application	Handles job applications CRUD operations and filtering.

Service	Responsibilities
Service	
Interview Service	Manages interviews and filtering by type or date.
Contact Service	Manages personal contacts related to job hunting.
Analytics Service	Aggregates data from other services and provides statistics/insights.
Dashboard Service	Composes and aggregates data from multiple services for the dashboard view.
Notification Service	Sends reminders for interviews or updates about applications.
Search Service	Provides search functionalities for users, applications, and contacts.
File Upload Service	Handles resume uploads and media storage.
API Gateway	Entry point for client requests; routes them to the appropriate services.
Auth Service	Handles login, registration, and token issuance.

Services and Details:

1. User Service

- **Responsibilities**: Profile management.
- **Communicates with**:
  - File Upload Service (for uploading resumes).
  - Contact Service (to cross-reference if a contact is a user).

2. Application Service

- **Responsibilities**: Add/edit job applications, filter/search by status, company, etc.
- **Communicates with**:
  - Analytics Service (via events).
  - Dashboard Service (REST/gRPC).
  - Interview Service (REST/gRPC).

3. Interview Service

- **Responsibilities**: Save and filter interviews.
- **Communicates with**:
  - Notification Service (for scheduling reminders).
  - Dashboard Service (for upcoming interviews).

- Application Service (REST/gRPC).

4. Contact Service

- **Responsibilities**: Manage user's professional contacts.
- **Cross-link**: Detect if a contact is a registered user (calls User Service).

5. Analytics Service

- **Responsibilities**: Aggregate app data into metrics and charts.
- **Consumes**: Events from Application Service, Interview Service.

6. Dashboard Service

- **Responsibilities**: Compose and aggregate data from other services.
- **Composition Pattern**: Makes synchronous REST or gRPC calls to:
  - Application Service
  - Interview Service

7. Notification Service

- **Responsibilities**: Notify users about upcoming interviews, application updates.
- **Delivers via**: Email, SMS, or in-app notification queue.

8. Search Service

- **Responsibilities**: Full-text search for applications, users, and contacts.

9. File Upload Service

- **Responsibilities**: Handle resume uploads.
- **Communicates with**:
  - User Service (to attach uploaded file to profile).

10. API Gateway

- **Responsibilities**: Routes external requests, handles auth, rate limiting, and maybe GraphQL.

11. Authentication Service

- Handles:
  - OAuth2 / JWT Token issuing

- Login/logout

Tech Stack:

- Backend: Node.js, Spring Boot and Django.
- Message Queue: Kafka / RabbitMQ.
- **API Gateway**: Kong / Express Gateway / GraphQL.
- **Database**: PostgreSQL / MongoDB.
- **Search**: Elasticsearch / Meilisearch.
- **Storage**: AWS S3 for resumes.
- **Auth**: JWT or OAuth2.
- **CI/CD**: GitHub Actions / GitLab CI / Docker/ Jenkins.