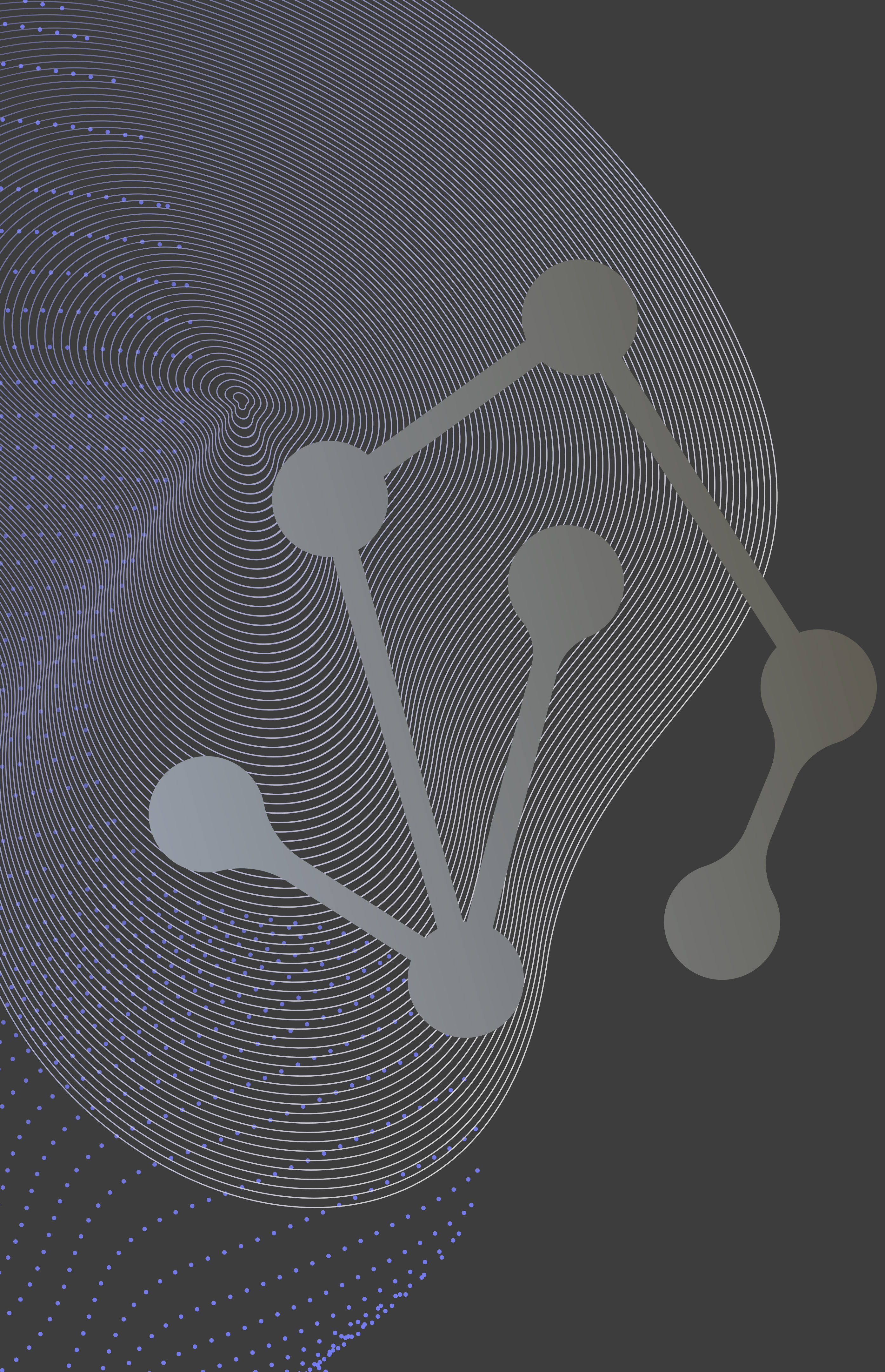
RESTful Backend Overview

Skills Marketplace API

Team members

- Enzo
- Grace
- Khalid
- Nickitta



Agenda

Overview of project sections

01 Introduction

02 Introduction to the Hiring Platform API

03 API Structure & Conventions

04 Endpoint

05 Authentication

06 Summary & Question

Introduction to the Hiring Platform API

- A RESTful web API designed for managing users, job listings, skills, and applications.

- Useful for job portals, HR platforms, or recruitment services.

- Authentication is required for most operations.
- Follows common HTTP verbs:
- GET, POST, PATCH, DELETE

- Key Features:
- User management
- Job posting & applications
- Skill tagging
- JWT-based security

API Structure & Conventions

- Base URL: Provided in the docs.
- RESTful resource-based routing:
  - /users, /jobs, /skills, /applications
- Common HTTP Status Codes:
  - 200 OK, 201 Created, 400 Bad Request, 401 Unauthorised, 404 Not Found
- Relationships:
  - Users ⇔ Jobs (users post jobs)
  - Jobs ⇔ Skills (jobs require specific skills)
  - Users ⇔ Applications ⇔ Jobs

Users Endpoint

- URL: /users
- GET: List all users
- POST: Create a new user
- PATCH: Update user info (e.g. name, email)
- DELETE: Remove user (admin only)

```
POST /users
{
  "name": "Khalid",
  "email": "khalid@example.com",
  "password": "securePass123"
}
```

Skills Endpoint

Routes:

- GET /skills – Retrieve list of skills.
- POST /skills – Create a new skill.
- PATCH /skills/:id – Update skill details.
- DELETE /skills/:id – Delete a skill.
- Access: Authenticated actions only.

Jobs Endpoint

- URL: /jobs
- GET: List jobs (with filtering/sorting)
- POST: Create a job (user must be authenticated)
- PATCH/DELETE: Update or remove own jobs

```
POST /jobs
{
  "title": "Frontend Developer",
  "description": "React and Tailwind experience required.",
  "skills": ["JavaScript", "React"],
  "deadline": "2025-06-30"
}
```

Applications Endpoint

Routes:

- POST /applications – Apply to a job.
- GET /applications?userId=123 – Retrieve applications by user.
- PATCH: /applications/:id/-Update an application.
- DELETE: /applications/:id/-Delete an application.
- Note: Only the applicant or an admin can update/delete applications.

Authentication

- Uses JWT (JSON Web Token) for securing APIs.
- Token sent in the Authorisation header:
- Bearer <token>
- Required for most actions (like posting jobs, applying).
- Returns 401 Unauthorised if missing or invalid.

Example

Authorisation: Bearer <your_token_here>

Without token → 401 Unauthorized

Error Handling

- Standard HTTP Status Codes:
  - 200 OK – Successful request.
  - 201 Created – Resource successfully created.
  - 400 Bad Request – Invalid request parameters.
  - 401 Unauthorised – Authentication required or failed.
  - 404 Not Found – Resource not found.
  - 500 Internal Server Error – Server encountered an error.

Summary

- Highlights:
    - RESTful API design with clear resource-based endpoints.
    - Secure access through JWT authentication.
    - Comprehensive management of users, skills, jobs, and applications.
    - Flexible querying with support for filtering, sorting, and pagination.

Q&A