**Rubric for Job Application Tracker Hub**

Phase 1: Planning and Design

**1. Define Project Scope**

* **Goal**: Clearly articulate the problem you're solving and the features you’re building.
* **Tasks**:
  + Write a project description (e.g., “A hub for managing job applications, trends, and AI-powered recommendations.”).
  + List core features: user management, dynamic dashboards, AI insights, etc.
* **Deliverable**: Finalized project scope document.
* **Timeframe**: 1 day.

**2. Plan the System Architecture**

* **Goal**: Design the high-level structure of your project.
* **Tasks**:
  + Define frontend-backend communication.
  + Choose technology stack (e.g., React for frontend, Django for backend, PostgreSQL for the database).
  + Create an architecture diagram.
* **Deliverable**: Completed architecture diagram.
* **Timeframe**: 2 days.

Phase 2: Backend Development

**3. Database Schema Design**

* **Goal**: Build the database to store users, applications, teams, etc.
* **Tasks**:
  + Create tables for users, applications, reminders, and AI insights.
  + Implement relationships (e.g., foreign keys for user ownership of applications).
* **Deliverable**: Fully functional database schema in PostgreSQL/MySQL.
* **Timeframe**: 3 days.

**4. API Development**

* **Goal**: Create APIs for data management.
* **Tasks**:
  + Develop endpoints:
    - /signup and /login for authentication.
    - /applications for CRUD operations on applications.
    - /reminders for creating and listing reminders.
    - /dashboard/summary for stats visualization.
  + Test each endpoint with tools like Postman.
* **Deliverable**: Tested API server.
* **Timeframe**: 5 days.

**5. Integrate AI Models**

* **Goal**: Add AI-powered features for trends and recommendations.
* **Tasks**:
  + Build AI models (e.g., logistic regression for predictions).
  + Train models using dummy or historical data.
  + Expose AI results via endpoints (/ai/trends, /ai/recommendations).
* **Deliverable**: Functional AI API.
* **Timeframe**: 7 days.

Phase 3: Frontend Development

**6. Build Core User Interface**

* **Goal**: Create pages for the tracker.
* **Tasks**:
  + Develop signup/login forms.
  + Create application management pages with forms and tables.
  + Implement dynamic filters and search features.
* **Deliverable**: Core UI without advanced styling.
* **Timeframe**: 7 days.

**7. Add Dynamic Dashboards**

* **Goal**: Visualize data using graphs and charts.
* **Tasks**:
  + Use Chart.js or D3.js to add bar graphs, pie charts, etc.
  + Fetch data from the /dashboard/summary endpoint.
* **Deliverable**: Fully functional dashboard.
* **Timeframe**: 5 days.

**8. Design Collaboration Features**

* **Goal**: Enable shared trackers.
* **Tasks**:
  + Implement team creation and member management pages.
  + Allow team members to view and edit shared trackers.
* **Deliverable**: Collaborative tracker interface.
* **Timeframe**: 5 days.

Phase 4: Feature Enhancements

**9. Implement Reminders and Notifications**

* **Goal**: Send timely alerts to users.
* **Tasks**:
  + Schedule email reminders using Celery (or cron jobs).
  + Add in-app notifications for deadlines.
* **Deliverable**: Reminder and notification system.
* **Timeframe**: 5 days.

**10. Excel Import/Export**

* **Goal**: Allow users to import/export data.
* **Tasks**:
  + Parse Excel files on import, map columns to database fields.
  + Query database and format data for export.
* **Deliverable**: Excel import/export functionality.
* **Timeframe**: 3 days.

**11. Polish AI Features**

* **Goal**: Make AI recommendations user-friendly.
* **Tasks**:
  + Optimize AI model performance.
  + Add suggestions dynamically as users fill out forms.
* **Deliverable**: Integrated and enhanced AI features.
* **Timeframe**: 5 days.

Phase 5: Testing and Deployment

**12. Testing**

* **Goal**: Ensure the system is robust.
* **Tasks**:
  + Write unit tests for APIs and AI models.
  + Perform frontend testing (e.g., using Cypress).
  + Conduct integration testing for overall workflow.
* **Deliverable**: Tested system with minimal bugs.
* **Timeframe**: 5 days.

**13. Deployment**

* **Goal**: Deploy the project for live use.
* **Tasks**:
  + Set up backend and database on AWS/GCP.
  + Use Netlify or Vercel for frontend hosting.
  + Ensure secure connections with HTTPS.
* **Deliverable**: Deployed project with documentation.
* **Timeframe**: 5 days.

Phase 6: Post-Launch Improvements

**14. Gather User Feedback**

* **Goal**: Identify pain points.
* **Tasks**:
  + Create feedback forms for users.
  + Analyze feedback and prioritize fixes.
* **Deliverable**: Feedback summary document.
* **Timeframe**: 3 days.

**15. Enhance Features**

* **Goal**: Add new features or improve existing ones.
* **Tasks**:
  + Use feedback to improve AI insights, dashboards, etc.
  + Optimize performance for large datasets.
* **Deliverable**: Feature-enhanced system.
* **Timeframe**: Ongoing.