



# Sprint Plan (15 Days Total)



## Week 1 – Project Setup + Core Models

### Day 1: Project Setup

- Install Python, pip, PostgreSQL/SQLite, Git, and VS Code.
- Create virtual environment & install Django.
- Create Django project: `django-admin startproject job_platform`.
- Create app: `python manage.py startapp jobapp`.
- Connect to PostgreSQL or SQLite.
- Run: `python manage.py runserver`.

### Day 2: User Authentication

- Create a CustomUser model (extend AbstractUser).
- Add `is_recruiter` boolean.
- Register user model in `settings.py`.
- Create user registration & login views.
- Create registration/login templates.

### Day 3: Admin Panel + Superuser

- Create a superuser.
- Register models in `admin.py`.
- Customize admin panel for managing users.
- Test admin login and dashboard access.

### Day 4: Profile + Resume Upload

- Create Profile model (One-to-One with User).
- Add fields: full name, contact, resume, profile pic.
- Create forms and templates to update profile.

- Use MEDIA\_URL to serve uploaded resumes.

#### Day 5: Job Posting System

- Create Job model with fields: title, company, location, description, etc.
- Build job creation form for recruiters.
- Display jobs on a “Job Listings” page.
- Restrict job creation to recruiters only.

#### Day 6: Job Application System

- Create Application model.
- Allow job seekers to apply with resume.
- Track status: Pending, Shortlisted, Rejected, Hired.
- Add “Apply” button to job detail page.

#### Day 7: Dashboard Views

- Create:
    - Job Seeker Dashboard (view apps/status).
    - Recruiter Dashboard (view posted jobs & applicants).
  - Add filters/search for job listings.
  - Add logic to show status updates on dashboard.
- 



### Week 2 – Video Interview + API + AI Tools

#### Day 8: Interview Scheduling

- Create Interview model.
- Recruiters can generate interview link (Jitsi or Twilio).
- Store link with time and job.
- Show link only to selected candidates.

#### Day 9: Email Notifications

- Set up SMTP in .env.

- Send email on: registration, apply, shortlisted, interview scheduled.
- Use Django signals to automate notifications.

#### Day 10: Google Login Integration

- Use social-auth-app-django.
- Enable Google OAuth login.
- Store GOOGLE\_CLIENT\_ID and SECRET in .env.

#### Day 11: API Endpoints (DRF)

- Install Django REST Framework.
- Create APIs:
  - Register/Login
  - Job CRUD
  - Apply Job
  - View Applications
  - Interview URL
- Use JWT authentication for secure access.

#### Day 12: AI Interview Feature (Optional Free Tools)

- Integrate simple AI service for:
  - Transcription (use Google Speech-to-Text or Whisper)
  - Summary (simple text analysis or use GPT-4All/Novita API)
- Attach results to Interview model.

#### Day 13: Testing

- Write unit tests: models, views, forms.
- Integration testing: job application flow.
- Test Interview URL access based on roles.
- Use python manage.py test.

#### Day 14: Deployment Setup

- Choose platform (Render/Heroku).
  - Set DEBUG=False, use gunicorn.
  - Add White noise for static files.
  - Run: collect static, migrate.
  - Add custom domain & SSL (optional).
- 

## **Day 15: Final Touches + Demo + Docs**

- Polish UI: CSS for all templates (login, register, job list, dashboard).
- Add README and project documentation.
- Record a demo video of app usage.
- Test end-to-end as:
  - Admin
  - Recruiter
  - Job Seeker