

# Django Developer Test Assignment

## Objective:

Build a simple **Job Application Management API** using **Django REST Framework**. This will test your understanding of models, serializers, APIs, and validation.

## 1. Models

### Applicant

- id: AutoField (Primary Key)
- name: CharField(max\_length=100)
- email: EmailField(unique=True)
- phone: CharField(max\_length=15, optional)
- resume: FileField(upload\_to='resumes/', optional)
- applied\_on: DateTimeField(auto\_now\_add=True)

### Job

- id: AutoField (Primary Key)
- title: CharField(max\_length=100)
- description: TextField
- posted\_on: DateTimeField(auto\_now\_add=True)

### Application

- id: AutoField (Primary Key)
- applicant: ForeignKey(Applicant, on\_delete=CASCADE)
- job: ForeignKey(Job, on\_delete=CASCADE)
- status: ChoiceField(['applied', 'shortlisted', 'rejected'], default='applied')
- applied\_on: DateTimeField(auto\_now\_add=True)

## 2. API Endpoints

Endpoint	Method	Description
/api/applicants/	GET, POST	List or create applicants
/api/applicants/{id}/	GET, PUT, DELETE	Retrieve, update or delete applicant
/api/jobs/	GET, POST	List or create jobs
/api/jobs/{id}/	GET, PUT, DELETE	Retrieve, update or delete job
/api/apply/	POST	Apply for a job
/api/applications/	GET	List all applications
/api/applications/{id}/	PATCH	Update application status

## 3. Additional Requirements

- Prevent duplicate applications (same applicant cannot apply twice for the same job).
- Use ModelSerializer with validation.
- Return proper HTTP status codes.
- Add search filter on applicants by name or email.

## 4. Bonus (Optional)

- Add JWT authentication (SimpleJWT).
- Add pagination for list endpoints.
- Write unit tests for at least one API endpoint.

## 5. Submission

- Upload code to a public GitHub repository.
- Include setup instructions in README.md.
- Optional: add .env.example for environment variables.

## 6. Evaluation Criteria

Area	Description
Django Models	Proper relationships and constraints
API Design	RESTful, consistent structure
Code Quality	Clean, readable, DRY
Validation	Prevents bad data and duplicates
Testing	Basic API tests implemented
Bonus	JWT, pagination, filters