

To use **Python + Django + HTML/CSS/JavaScript**, here is a **combined and detailed roadmap** that aligns your feature set with the recommended monolithic architecture using Django.

Full Project Roadmap

- **Backend:** Python + Django
- **Frontend:** Django Templates + Bootstrap/Tailwind CSS + Vanilla JavaScript
- **Database:** PostgreSQL (prod) / SQLite (dev)
- **File Handling:** Local for dev, AWS S3 (prod)

Phase 1: Planning & Setup

1.1 Requirements Finalization (from PDF)

1. Job Management (CRUD)
2. Application System (Form + Resume Upload)
3. Screening Questions (Custom per job)
4. Admin Dashboard
5. Deadline Handling
6. Responsive Design
7. Filtering & Export
8. Auto-close on deadline
9. Secure Login

1.2 Environment Setup

Phase 2: Database Design & Models

2.1 Key Models

Model	Description
User	Admins & applicants (Django auth extended)
JobPost	Title, description, deadline, is_open
ScreeningQuestion	FK to JobPost, type, text
Application	FK to User & JobPost, timestamp
Answer	FK to Application & ScreeningQuestion
UploadedFile	Resume (PDF), Cover Letter (optional)

2.2 Model Features

1. JobPost: auto-close on deadline
2. Application: status tracking, validation

3. UploadedFile: max 10MB, PDF only

Phase 3: Backend Features (Django)

3.1 Job Management (Admin)

1. Create/edit/delete job posts
2. Set salary, location, and deadlines
3. Add custom screening questions per job

3.2 Application System

1. Public-facing form
2. Resume (PDF) + optional cover letter
3. Screening questions per job
4. Auto-save with localStorage + JS
5. Block application after deadline

3.3 Admin Dashboard

1. Job overview (open/closed)
2. Applications per job
3. Bulk actions (e.g., export, delete)
4. Stats on expired/expiring jobs
5. Extend/reopen jobs
6. Filters: name, email, phone, date, documents, question answers

Phase 4: Frontend Development

4.1 Public Pages

1. **Homepage:** List open jobs in responsive grid
 - ❖ Filter: open/closed
 - ❖ Job cards: title, salary, deadline
2. **Job Detail Page:** Description, requirements, “Apply” button

4.2 Application Form

1. Fields: Name, email, phone
2. Resume Upload (drag & drop)
3. Cover letter (optional)
4. Screening Questions (dynamic types: short, yes/no, MCQ)
5. Client-side validation + deadline check
6. Auto-save to localStorage

4.3 Admin Interface

1. Job Management UI
2. Application filters and sorters
3. Deadline controls
4. Export functionality (CSV)