

# **RESUME BUILDER DJANGO DOCUMENTATION**

## **Table of Contents**

1. Introduction
2. Installation
3. Project Structure
4. Configuration
5. Usage
- 6 Schema Design
7. API
8. Urls
9. Views
10. Features
11. Validations
12. Appendix and Additions

### **1. Introduction**

Resume Builder Django is a web application that allows users to create and manage their resumes easily. It is built using the Django web framework and provides a user-friendly interface for building professional resumes.

### **2. Installation**

To run the Resume Builder Django application, follow these steps:

#### **Prerequisites**

- Python 3.x
- pip

#### **Installation Steps**

1. Clone the repository:

```
git clone https://github.com/your-username/resume-builder-django.git
```

2. Navigate to the project directory:

```
cd resume-builder-django
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Apply database migrations:

```
python manage.py migrate
```

5. Create a superuser account:

```
python manage.py createsuperuser
```

6. Start the development server:

```
python manage.py runserver
```

7. Access the application at <http://localhost:8000/>

### 3. Project Structure

```
'''
resume_builder_django/
|
|— rev/
|   |— settings.py
|   |— urls.py
|   |— ...
|— revapi/
|   |— ...
|— requirements.txt
|— manage.py
|— venv
'''
```

- `**resume_builder/**`: Contains the Django project settings and URL configurations.
- `**manage.py**`: Django management script.

### 4. Configuration

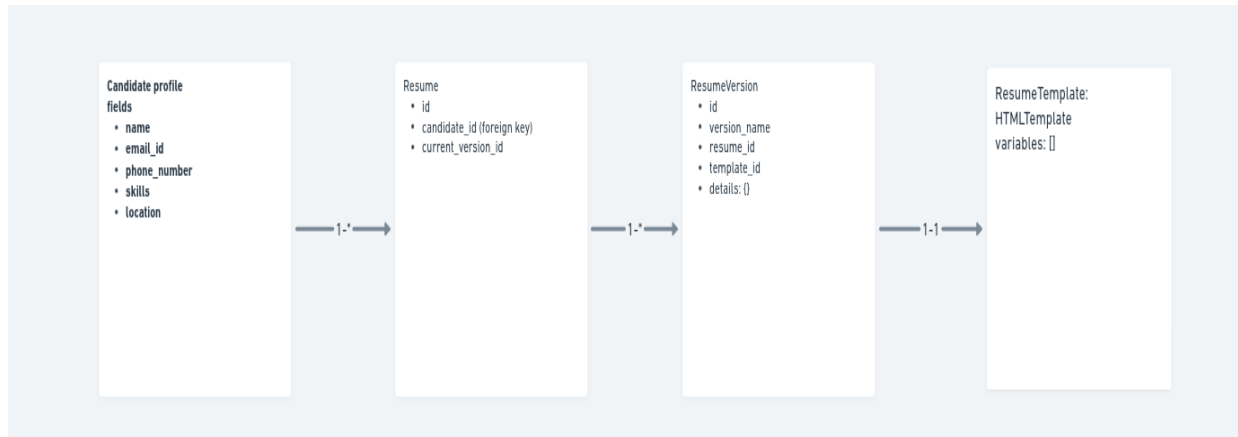
Update the following settings in the `settings.py` file:

- `DEBUG`: Set to `False` in production.
- `ALLOWED_HOSTS`: Add the hostnames or IP addresses allowed to serve the application.

### 5. Usage

1. Access the application at <http://localhost:8000/>
2. Give name input **Name**, **Address**, **Number**, **Email** , **Location** , **TechSkills**

### 6. Schema Design



There are two tables/models candidate and multi-candidates.

#### First Table

Name  
Address  
Number  
Email  
Location  
Experience  
Techskills

#### Second Table

Text field { Which takes json bulk data and maps in First Table }

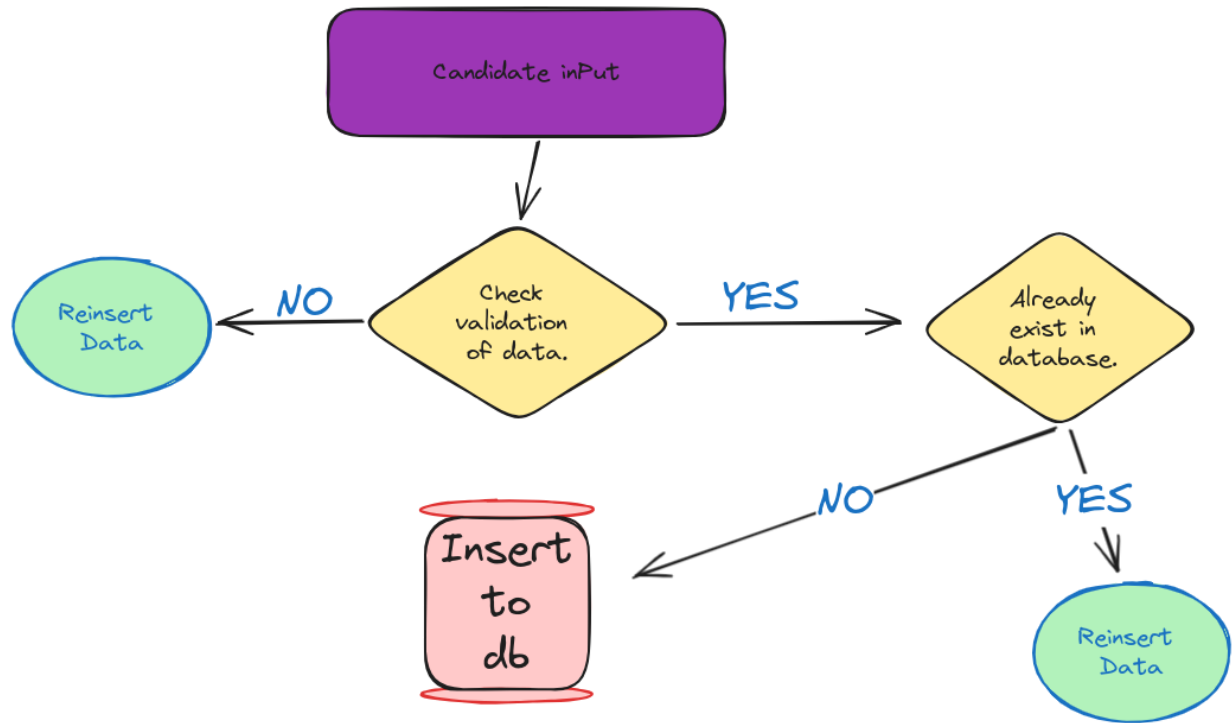
### 7. API

#### CandidateProfileViewSet

Check Validation For Candidates.

- Whether name is empty or not
- Address is valid or not
- Number is 10 digits or not and from which country.
- Email is valid or not.
- Location is Valid or not.

Insert it to the database and save the pdf locally.



### MultiCandidateViewSet

Only 1 text field is there which inserts resume data in bulk after validation.

## 8. Urls

[http://localhost:8000/candidate\\_info/](http://localhost:8000/candidate_info/)

After running the project in local and deploying at some server

This is the link that can be used to input your details and further those details

Helps to make pdf based resumes.

## 9. Views

Most of the views are CBV imported from *rest\_framework.generics*, and they allow the backend api to do the basic CRUD operations expected, and so they inherit from the *ListAPIView*, *CreateAPIView*, *RetrieveAPIView*, ..., and so on.

### CandidateProfileViewSet

~ Showcases all the resumes as given by the user through json.

~ Before Post checks whether the resume data is validated or not.

### MultiCandidateDataViewSet

```

~ Get all the data of multi resumes in json format.
~ Check validation for each json.
~ Provides the resume in pdf format.

```

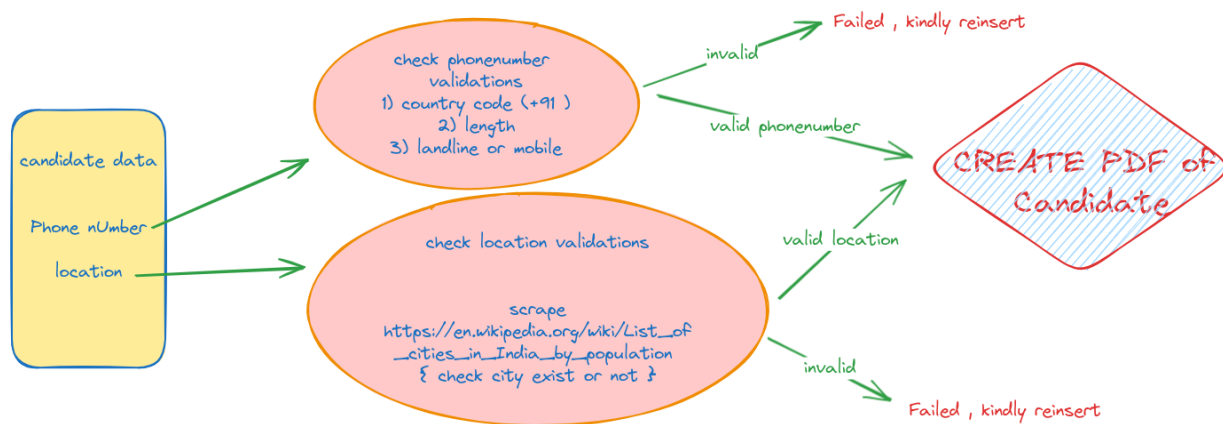
## 10. Features

- Dynamic resume creation with customizable sections.
- Multi-Validation Checks for resumes.
- No repeating Candidates for validation.
- Ability to download resumes in various formats (PDF).

## 11. Validations

Multiple validations are here that can be there in Resume Builder,

- ~ Checking location of candidate
- ~ Checking phone number validity of candidate
- ~ Checking if the same resume is being created.



## 12. Appendix and Additions

1. Addition of Image upload in api.
2. Addition for repeated resumes based on dates.
3. Multiple formats apart from pdf.
4. Number and Email Verification using OTP verification in Django.
5. Experience verification through HR linkedin.
6. Usage of Chatgpt for recommendation and updates.
7. Email to the candidate instead of downloading a resume to keep the data authenticity.
8. Mail Facility to various HRs if we have the database or emails of hrs.

