Part 1

This program uses Django Rest Framework which is a python framework.

# **Installation**

Install the latest version of python3.

Download the project files

Use pip to Install Django.

Cd the project folder.

Create a virtual environment.

Activate the virtual environment.

pip install -r requirements.txt

# Github link

<https://github.com/lenimbugua/focus>.

Part 2

1. **Which specifications were you unable to address in your submitted application?**

**-** Add/Edit Employee

* + Submit button is enabled if form is valid.
  + Include some type of activity indicator on submission
  + Display error messages from the backend

-Upload employees

* + Form should be displayed in a modal on top of home page.
  + Submit button is enabled if file has been chosen
  + Include some type of activity indicator on submission
  + Result should contain how many records succeeded and how many failed
  + Include cancel button
  + Display error messages from the backend

1. **If you had more time, how would you improve your application?**
2. **In the MVP, ACME allowed only one phone number per employee. Assume ACME would like to allow employees to have more than one phone number, such as home, mobile and work. How would you migrate the data and change schema to accommodate such a request?**

* Create a new table called phone.
* Include all the fields (home, mobile, work).
* Implement a primary key for this table.
* Implement a **one to many** relationship between the employees table and this table. through foreign keys. The phone field on the employees table would be a foreign key. referencing the primary key on the phone table.
* Implement foreign key constraints such us ON-DELETE\_CASCADE.
* Perform migrations.

1. **In the MVP, the departments for ACME were assumed to be deterministic. Assume ACME would like to keep the departments open-ended. For example, the CSV upload for new employees can contain new departments not already used or the user might enter a new department in the add user form. How would you migrate the data and change the schema? Secondly, how you would change the add user(s) backend code to fit this new schema? (There’s no need to write code or pseudocode, explain in words what steps the code should take).**

* Create a new table for the departments
* Include a primary key and the names of the departments as the fields of this table.
* Implement a **one to one** relationship between the employees table and this table. through foreign keys. The department field on the employees table would be a foreign key referencing the primary key on the department table.
* Implement foreign key constraints such us ON-DELETE\_CASCADE.
* Perform migrations.
* At the backend during upload, perform a check to determine if the department already exists.
* If it does exist perform normal upload.
* If it doesn’t exist create a new department, populate the department table with its value and perform upload.

1. **ACME has grown to a large company with over 10,000 employees. They have noticed that downloading the entire CSV file takes a long time. At times, the front end seems frozen as the request takes a long time to build and serve the CSV file. How would you solve this problem?**

* I would use asynchronous programming on the frontend.
* Downloading would be done on the background.
* Implement a notification system to notify when the download is complete.

1. **Which part of the assignment did you enjoy the most?**

Designing the user interface

1. **Which part of the assignment did you enjoy the least?**

Wiring the backend to frontend