SIX WEEK INTERNSHIP TRAINING REPORT ON

Web development using python

( django framework)

Completed at

NIWE, Chennai



OWD, DAF & IT division,

National Institute of Wind Energy, Chennai.

Submitted To: Shri.A.G.Rangaraj Submitted by: Kaviya Ns

Dy.Director (F&IT)

Table of Contents: page no.

1. Abstract ----------------------------------- 3
2. Objectives ----------------------------------- 4
3. What I learnt ----------------------------------- 4
4. Introduction of the organisation ------------------------------------ 4
5. Project modules ------------------------------------ 5
6. Project overview ------------------------------------ 6

Objectives

Technologies used

1. Project implementation ------------------------------------ 7

Form creation

Storing form data

Dynamic Fetching to excel template

1. Project challenges and solution ------------------------------------- 11

Customizing excel template

Handling large data sets

1. Conclusion -------------------------------------- 12

**ABSTRACT:**

This internship report focuses on a Python Django project that involved the creation of dynamic forms, storing form data in a MySQL database, and dynamically fetching the data to an Excel template using configuration parameters, implementation of multiple models and the establishment of foreign key relationships. The report provides an overview of the project's objectives, technologies used, and implementation details. It highlights the challenges encountered during the project and the solutions employed to overcome them.

The form creation feature was implemented using Django's form handling capabilities, allowing users to create custom forms dynamically. The form data was stored in a MySQL database for persistence, utilizing Django's ORM for efficient database interaction. To dynamically populate an Excel template with the stored data, configuration parameters were utilized for customization. The pandas and openpyxl libraries were employed for data manipulation and export.

Throughout the internship, various challenges were faced, including dynamically rendering forms and exporting data to an Excel template. However, these challenges were successfully addressed by leveraging Django's form rendering capabilities and utilizing pandas and openpyxl for data manipulation and export.

Overall, the internship project provided valuable practical experience in web application development, database integration, and data manipulation. It enhanced the skills in Python programming, Django framework, and working with libraries such as pandas and openpyxl. The report concludes by emphasizing the significance of the internship project in terms of the professional growth and its contribution to the understanding of real-world application development.

**OBJECTIVES:**

 To demonstrate my knowledge and learn more.

 How to work in company with seniors and in team.

 Learn the things in the company respected individual should know.

 Deploy my ability, work professionally, capacity to work, my speaking skills, to know the work.

**WHAT I LEARNT:**

 Introduction of python

 Sequences and operations

 Deep Dive – Functions, Sorting, Errors and Exception, Regular Expressions and Packages

 Object Oriented Programming in Python

 Debugging

 Introduction to Django Web Framework

 Templates and Forms

 Database connectivity

 Excel template and excel configuration

 Working with multiple models

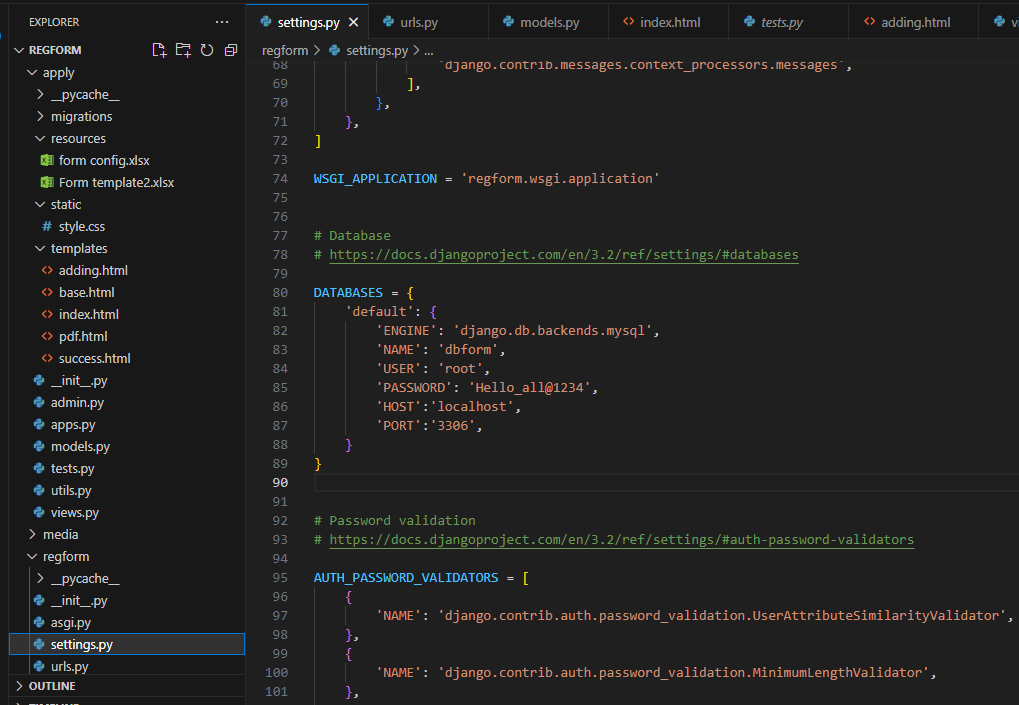
**INTRODUCTION OF THE ORGANISATION**:

The National Institute of Wind Energy (NIWE) is an autonomous research and development institute located in Chennai, India. Established in 1998 under the Ministry of New and Renewable Energy (MNRE), NIWE is dedicated to promoting the development and utilization of wind energy in India.

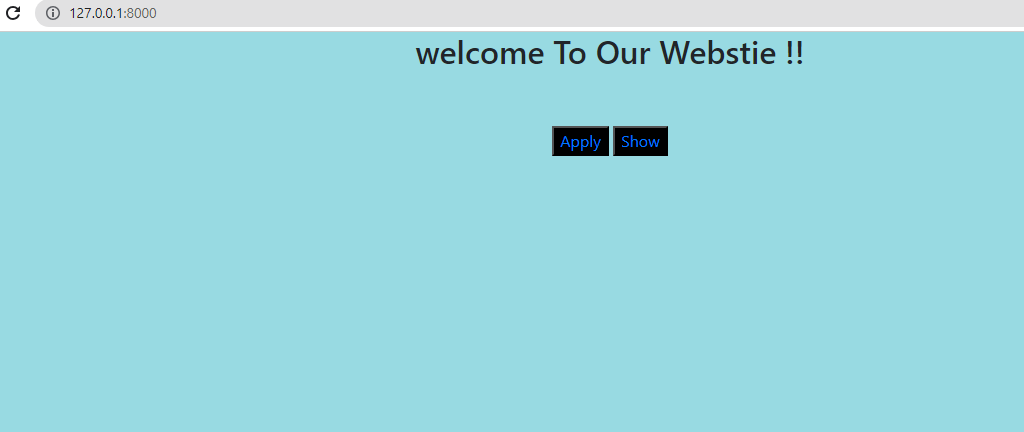
NIWE serves as a leading research and development institute in the field of wind energy in India. Through its comprehensive research, testing, capacity building, and policy support initiatives, NIWE plays a vital role in fostering the growth and integration of wind power, contributing to India's renewable energy goals and the global transition towards clean and sustainable energy sources.

**PROJECT MODULES:**

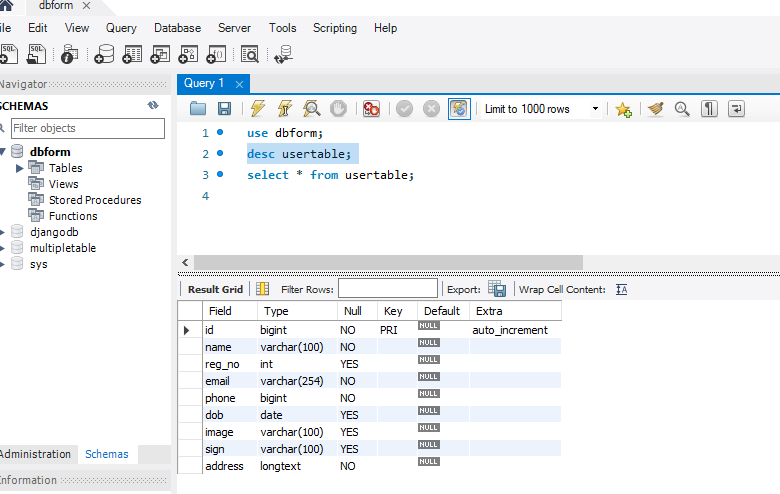
1. **Django** : Django is a high-level Python web framework that provides a robust set of tools and features for building web applications. It offers built-in support for form handling, database integration, and URL routing.
2. **MySQL Connector:** MySQL Connector is a Python module that allows Python applications to connect and interact with MySQL databases. It provides an interface to execute SQL queries, retrieve data, and manage database transactions.



***My Index page:***

******

***Table description:***



1. **Pandas:** Pandas is a powerful data manipulation and analysis library in Python. It provides data structures like DataFrame that facilitate easy manipulation and transformation of tabular data. Pandas can be used to convert retrieved data from the MySQL database into a DataFrame for further processing.
2. **openpyxl:** openpyxl is a Python library for reading and writing Excel files. It provides functionality to create, modify, and extract data from Excel spread sheets. openpyxl can be used to dynamically populate an Excel template with the retrieved data and configure its formatting.
3. **Django Form:** Django's built-in form handling capabilities can be utilized to create dynamic forms. It offers various form field types and validation options, making it easy to generate and process user-submitted form data.
4. **Django ORM:** Django's Object-Relational Mapping (ORM) allows seamless interaction with the MySQL database. It provides a high-level API to define and query database models, simplifying data storage and retrieval operations.
5. **Django Views:** Django views are Python functions or classes that handle HTTP requests and generate HTTP responses. Views can be implemented to retrieve data from the database, process it, and generate dynamic Excel files using the configuration parameters.
6. **Django Templates:** Django templates enable the creation of HTML files with placeholders for dynamic content. Templates can be used to generate user interfaces, display form data, and provide the option to download the generated Excel file.

**PROJECT OVERVIEW:**

***Objectives:***

The primary objectives of the project are as follows:

* Utilize Python Django framework to build a web application.
* Implement a form creation feature enabling users to generate dynamic forms.
* Implement multiple models to represent forms and their associated fields.
* Establish foreign key relationships between the models to maintain the form-field associations.
* Store the submitted form data in a MySQL database for persistence.
* Retrieve the stored data and populate an Excel template dynamically, utilizing configuration parameters for customization.

***Technologies Used:***

The project employed the following technologies:

* Python
* Django
* MySQL
* Pandas
* Openpyxl

**PROJECT IMPLEMENTATION:**

***Form Creation :***

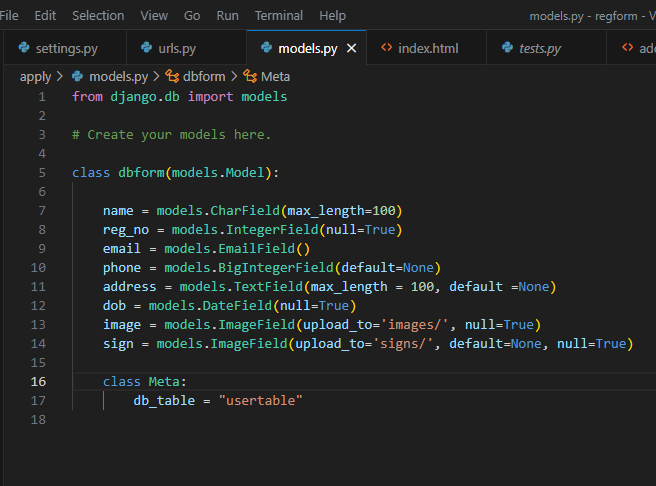
To facilitate form creation, Django's form handling capabilities were leveraged. A model representing the form structure was created, incorporating fields such as form name, field names, field types, and validation rules. Users could dynamically generate forms by adding and configuring fields through a user-friendly interface.

* ***Create a django model:***

Define a django model that represents the structure of the data to be exported.

Each field in the model corresponds to a field in the form.

***My model:***

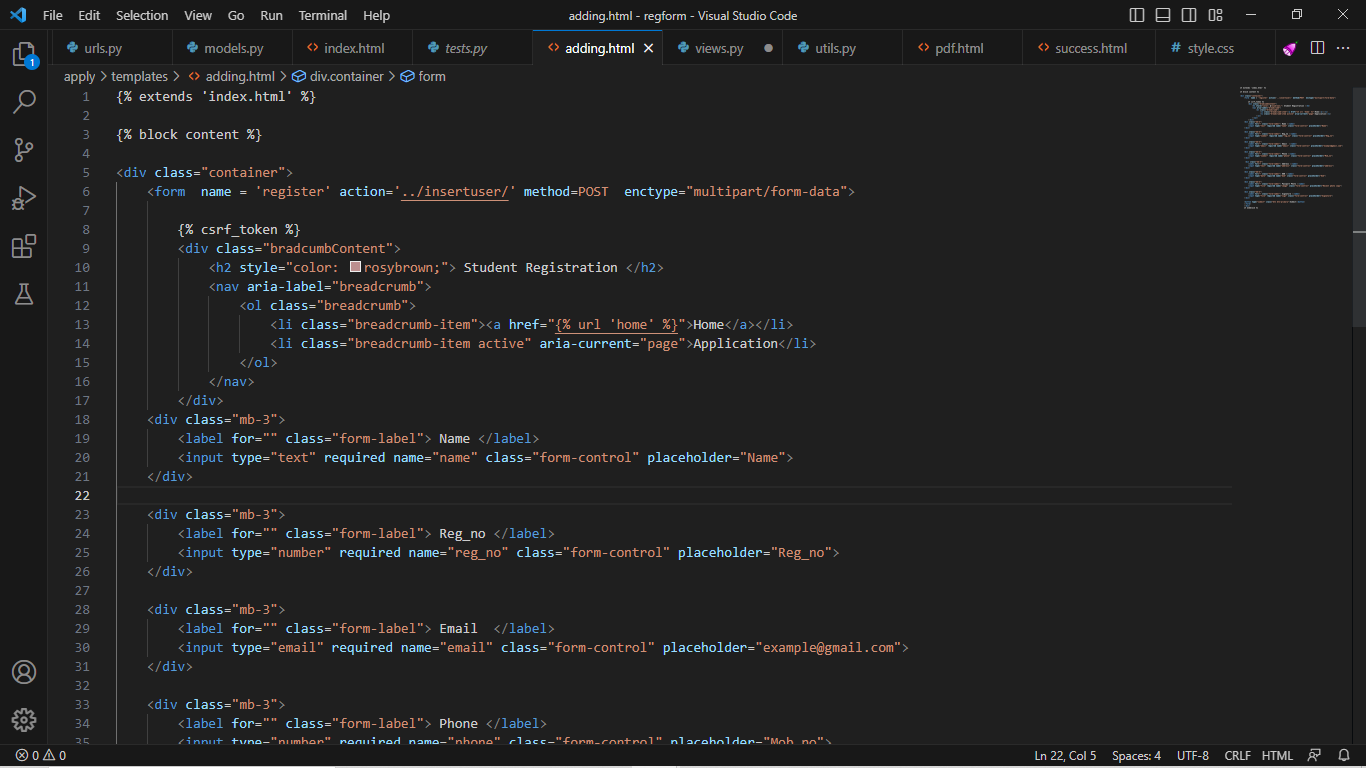


* ***Create a django form:***

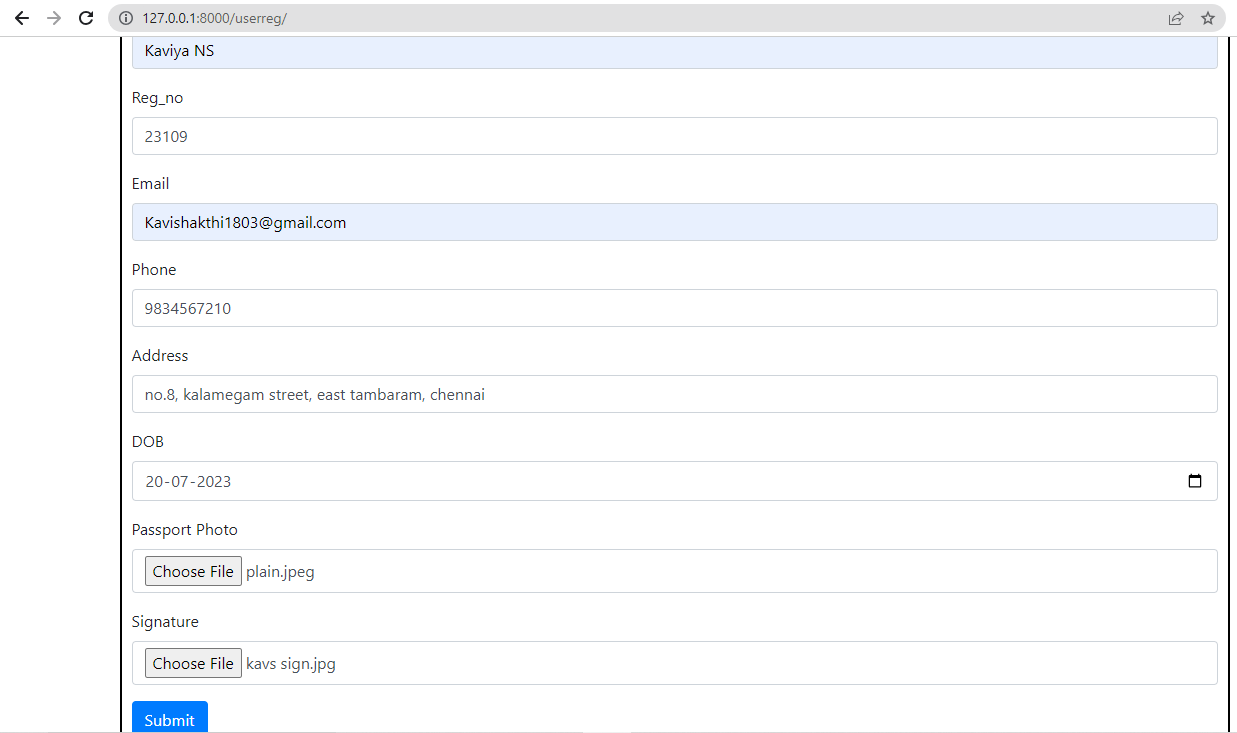
Define a django form based on the model created.

Add form fields that match the fields in the model.

***Actual code for form:***

******

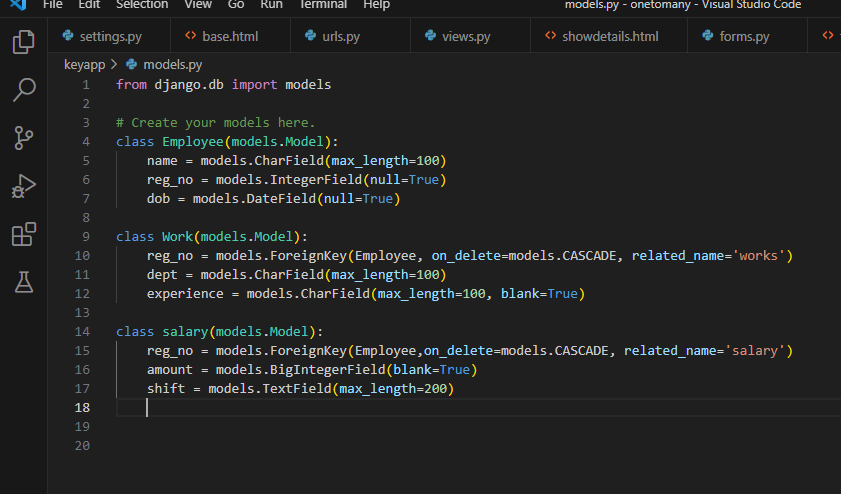
***Output:***

******

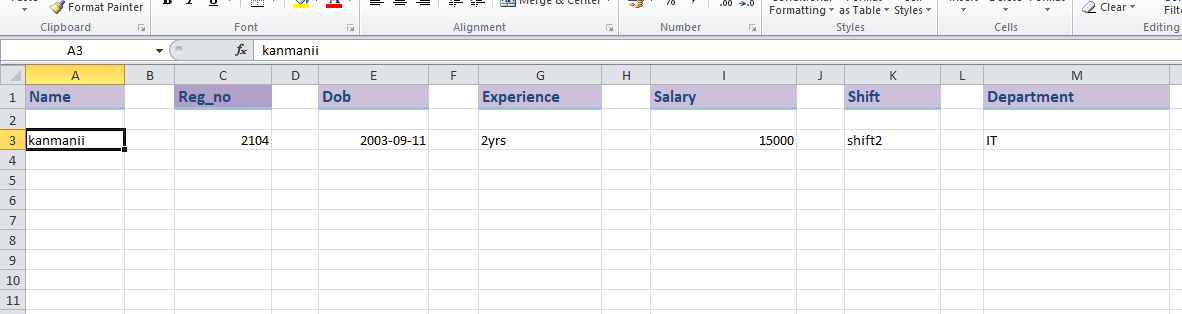
***Establishment of Foreign Key Relationships:***

Foreign key relationships were established between the FormModel and FieldModel using Django's ForeignKey field. This allowed each field to be associated with a specific form. The foreign key relationship maintained the integrity and coherence of the data structure, ensuring that fields were linked to the correct form.

***Multiple models:***

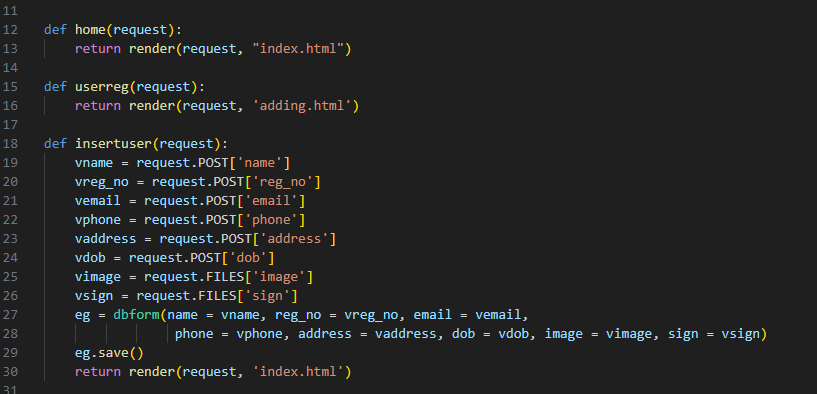
******

***Output of multiple models using foreign key:***

******

***Storing Form Data:***

Form data was stored in a MySQL database using Django's built-in ORM (Object-Relational Mapping). Whenever a form was submitted, the data was validated and saved to the corresponding tables in the database. This ensured data integrity and allowed for easy retrieval.

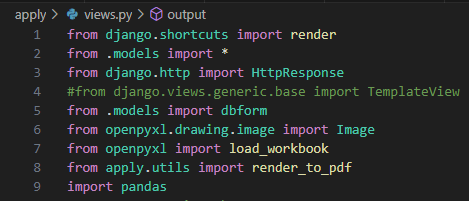
******

***Dynamic Fetching to Excel Template:***

To fetch stored form data and populate an Excel template dynamically, configuration parameters were employed. The project utilized pandas and openpyxl libraries in Python. A Django view was created to query the database and retrieve the desired form data. The retrieved data was then transformed into a pandas DataFrame, allowing for efficient data manipulation. Configuration parameters were used to customize the data population process and match the desired Excel template.

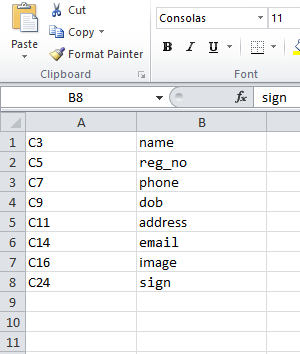
***Steps to Export Data from MySQL Database to Excel Template:***

**Step 1:** Install Required Packages and import them in the project views.py file. Install openpyxl library in the django environment using the command **‘pip install openpyxl’.**

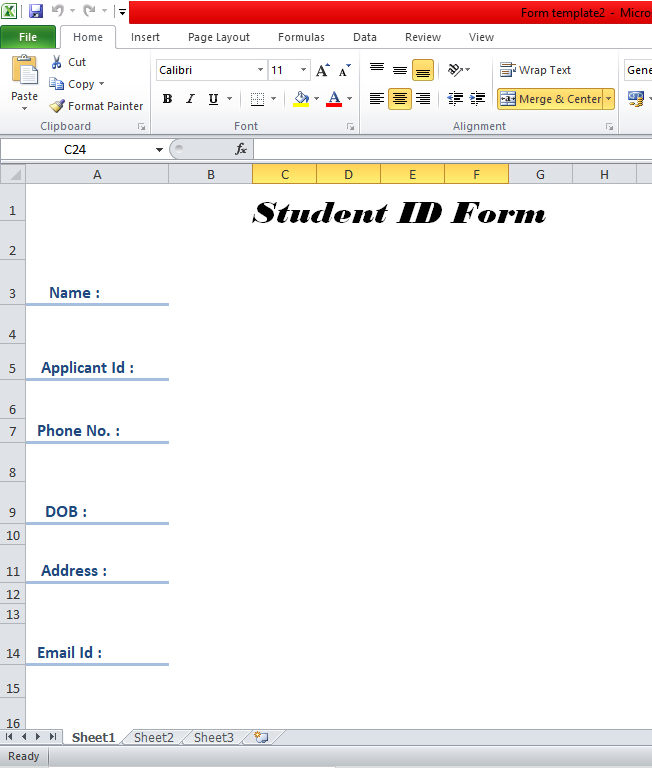


**Step 2:** Prepare the Excel Template. Create an Excel template that represents the desired structure and formatting for the exported data. I used the Microsoft Excel to create the template. Create a resource folder in the django project and save the template in that folder. Write the configuration parameters for exporting data and save it the same folder.

***My config excel sheet:***

******

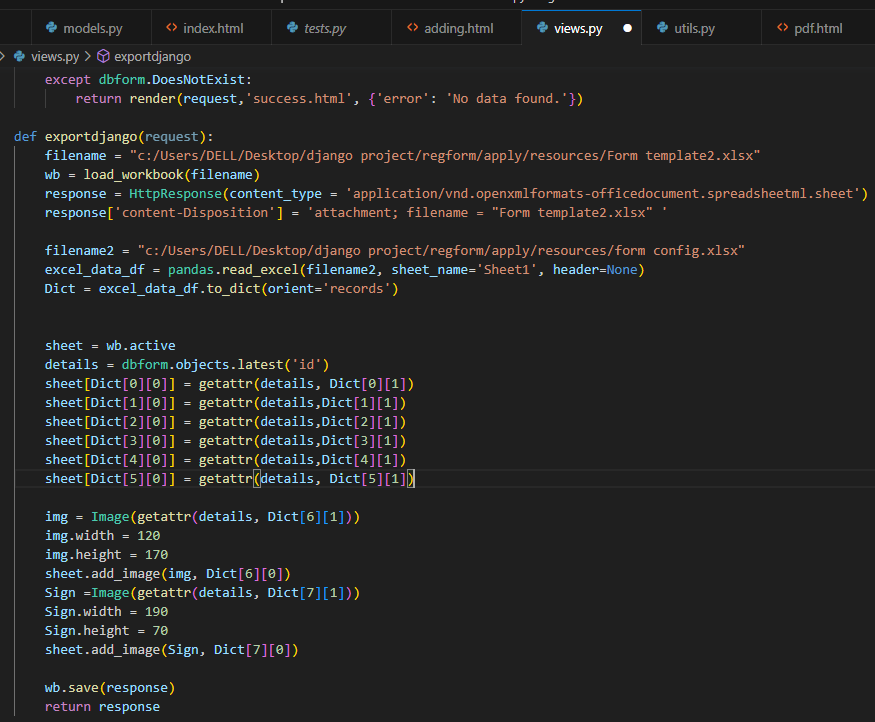
***My template:***

******

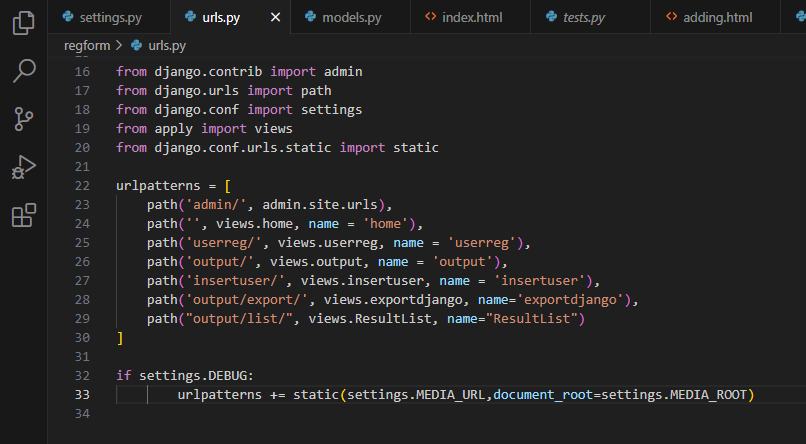
***Step3:*** specify the template path and define a view that will handle the export functionality. The view should perform the following tasks:

* Load the Excel template using the **openpyxl** library.
* Connect to the MySQL database using the configured parameters.
* Execute a query to retrieve the required data from the database.
* Populate the Excel template with the retrieved data.
* Close the database connection.
* Prepare an HTTP response containing the generated Excel file.

***Export django function:***

******

***Step 4:*** Define a URL Pattern In the Django application's **urls.py** file, define a URL pattern that maps to the view created in the previous step. This allows users to access the export functionality by visiting the defined URL.

******

**PROJECT CHALLENGES AND SOLUTIONS:**

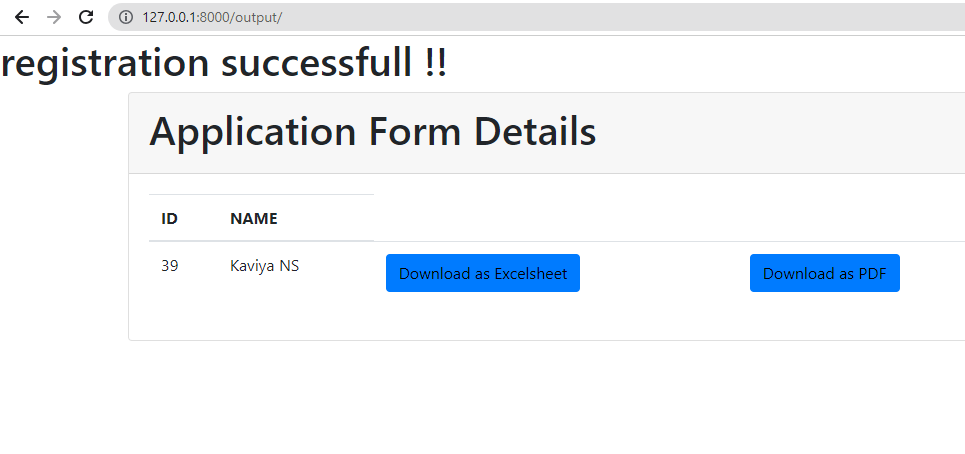
***Customizing Excel Template:***

One of the challenges faced was customizing the Excel template dynamically based on the configuration parameters. To address this challenge, the projects used openpyxl’s functionality to access and modify the template, ensuring the retrieved data was populated accurately.

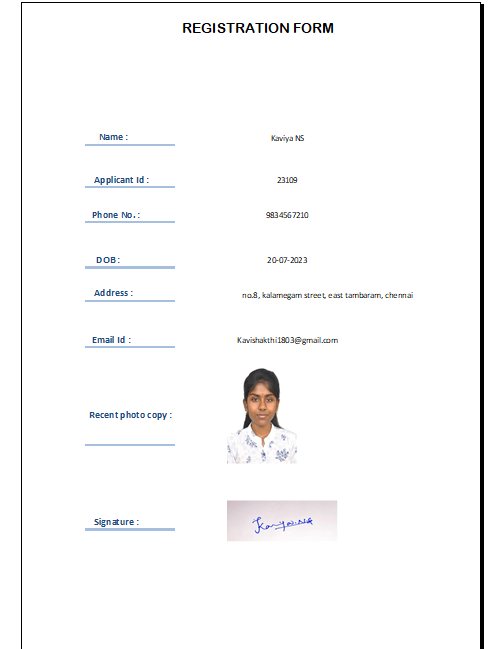
***Handling Large Data Sets:***

Another challenge encountered was efficiently handling large data sets when fetching data from the database. To overcome this challenge, the project utilized pagination techniques and optimized database queries to retrieve data in smaller chunks, ensuring performance and memory efficiency.

***Final output:***

******

After clicking download as excel sheet the below form will be downloaded.

******

**CONCLUSION:**

This internship project successfully accomplished the objectives of working with multiple models and establishing foreign key relationships in a Python Django application, form creation, storing form data in a MySQL database, and dynamically fetching the data to an Excel template using configuration parameters. The establishment of foreign key relationships ensured proper linkage between forms and their respective fields.

The project provided practical experience in working with Django's model relationships and database integration. It showcased the importance of maintaining data integrity and coherence by utilizing foreign keys to establish meaningful associations between models.

The internship project significantly enhanced the understanding of database modelling, relationship management, and dynamic data retrieval. It also expanded their knowledge and skills in Python programming, Django framework, and the utilization of libraries such as pandas and openpyxl for data manipulation and Excel generation.

In conclusion, the project's implementation of multiple models and foreign key relationships added a structured and organized approach to form creation, data storage, and dynamic Excel generation. The experience gained during the project will be valuable for future endeavours involving complex data relationships and management within Django projects.

PYTHON DJANGO

COMPLETED AT

YCT Academy, BHOPAL

PYTHON DJANGO

COMPLETED AT

YCT Academy, BHOPAL