VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

A Mini Project Report

On

"AUTOMATIC CERTIFICATE GENERATOR"

Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering in Computer Science and Engineering

Submitted by

Abhishek YS (1VE21CS007)
Aishwarya S (1VE21CS008)
Dharshan R (1VE21CS043)

Under the Guidance of
Mrs. Arpitha J
Asst. Prof, Dept. of CSE

Accredited by NAAC & NBA*
SVCE BENGALURU
SRI VENKATESHWARA COLLEGE OF ENGINEERING
— Affiliated to VTU, Approved by AICTE, Recognised by UGC u/s 2(f) & 12(B)—

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESHWARA COLLEGE OF ENGINEERING

Affiliated to VTU Belgaum & Approved by AICTE New Delhi) an ISO 9001:2008 Certified, Kempe Gowda International Airport road, Vidyanagar, Bengaluru, Karnataka, India-562157

2023– 2024

SRI VENKATESHWARA COLLEGE OF ENGINEERING
Vidyanagar, Bengaluru, Karnataka, India-562157

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Mini Project entitled **“AUTOMATIC CERTIFICATE GENERATOR”** is submitted by **Abhishek Y S [1VE21CS007], Aishwarya S [1VE21CS008], Dharshan R [1VE21CS043],** on partial fulfillment of sixth semester, Bachelor of Engineering in Computer Science and Engineering, Visvesvaraya Technological University for the academic year 2023-2024.

………………………………..	……………………………..
Signature of Course Teacher	**Signature of HOD**
Mrs. Arpitha J	**Dr. Hema M S**
Asst. Prof, Dept. of CSE	**Prof. & HOD, Dept. of CSE**

ABSTRACT

This Certificate Generator project automates the creation of personalized certificates, streamlining a traditionally manual and error-prone process. By reading recipient names from an Excel sheet, overlaying them onto a pre-designed template with OpenCV, and converting the images into PDFs using FPDF, the system ensures efficiency, accuracy, and consistency. Managed through a Django web application, the project offers a user-friendly interface for generating and storing certificates in a database. Ideal for educational institutions and events, this solution significantly reduces manual effort, ensures uniformity, and provides scalability. Future enhancements include template management, advanced customization, and integration with third-party services.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise, does not depend solely on individual efforts but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends. We would like to take this opportunity to thank them all.

First and foremost, we would like to express our gratitude to **Dr. Nageswara Guptha M**, Principal, SVCE Bangalore, who has always been a great source of inspiration.

We express our sincere regards and thanks to **Dr. Hema M S**, Professor and Head of the Department of Computer Science and Engineering, SVCE Bangalore, for her encouragement and support.

We are grateful to acknowledge the guidance and encouragement given to us by **Mrs. Arpitha J,** Assistant Professor, Department of Computer Science and Engineering, SVCE Bangalore, as the project coordinator and guide, who have rendered valuable assistance.

We also extend our thanks to the entire faculty of the Department of Computer Science and Engineering, SVCE Bangalore, who have encouraged us throughout the course of the project work.

Last but not least, we would like to thank our family and friends for their inputs to improve the project.

Abhishek Y S [1VE21VS007]

Aishwarya S [1VE21CS008]

Dharshan R [1VE21CS043]

TABLE OF CONTENTS

SL NO.	Title	Page No.
1	INTRODUCTION	1
2	PROJECT SETUP	2
3	IMPLEMENTATION	3
4	DETAILED EXPLANATATION	7
5	ADVANTAGES AND DISADVANTAGES	8
6	FUTURE ENHANCEMENT	10
7	RESULTS	11
8	CONCLUSION	12
9	REFERENCES	13

Chapter 1

INTRODUCTION

The Certificate Generator project aims to streamline the process of creating personalized certificates. Traditionally, generating certificates manually for each recipient can be time-consuming and prone to errors. This project automates the entire workflow by reading recipient names from an Excel sheet, overlaying these names onto a pre-designed certificate template, converting the customized templates into PDF format, and storing the resulting certificates in a database. This automation ensures efficiency, accuracy, and consistency in certificate generation.

The project employs the Django web framework for handling HTTP requests and database operations, OpenCV for image processing, and FPDF for PDF creation. By leveraging these technologies, the Certificate Generator can efficiently produce a large number of certificates, making it an ideal solution for educational institutions, training programs, conferences, and other events where certificates of participation or achievement are required.

1.1 Objectives

The primary objectives of the Certificate Generator project are:

1. **Automation of Certificate Creation** :- To automate the process of generating personalized certificates, reducing manual effort and human errors.

2. **Efficient Data Handling** :- To read and process recipient data from an Excel sheet, ensuring a streamlined workflow for certificate generation.

3. **Customizable Templates** :- To overlay recipient names onto a pre-designed certificate template accurately, allowing for easy customization and branding.

4. **PDF Generation** :- To convert the customized certificate images into PDF format, ensuring compatibility and ease of distribution.

5. **Database Management** :- To store the generated certificates in a database, enabling efficient record-keeping and retrieval.

6. **User-Friendly Interface** :- To provide a simple and intuitive interface for users to initiate the certificate generation process through a web application.

Chapter 2

PROJECT SETUP

2.1. Requirements

- Python 3.x
- Django
- OpenCV (cv2)
- Pandas
- FPDF
- Pillow (for handling image fields in Django)

2.2. Installation

1. **Install Django**:

   pip install django
2. **Install OpenCV**:

   pip install opencv-python
3. **Install Pandas**:

   pip install pandas
4. **Install FPDF**:

   pip install fpdf
5. **Install Pillow**:

   pip install pillow

2.3. Project Structure

- certificate_generator/: Root directory of the Django project.
- generator/: Django app responsible for certificate generation.
- views.py: Contains the logic for generating certificates.
- models.py: Defines the database model for storing certificate information.
- urls.py: URL configurations for the app.

Chapter 3

IMPLEMENTATION

3.1. Views

File: views.py

Python Copy code

```
import cv2
import pandas as pd
import os
from fpdf import FPDF
from django.shortcuts import HttpResponse
from .models import Certificate
def generate_certificates(request):
    # Read the Excel sheet
    data =
pd.read_excel('C:/Users/abhis/OneDrive/Desktop/ASQW/certificate_generator/generator/LIST1.xlsx')
    # Assuming your name column is named 'Name' in the sheet
    list_names = data['Name'].tolist()
    # Get the dimensions of the input template image
    template =
cv2.imread('C:/Users/abhis/OneDrive/Desktop/ASQW/certificate_generator/generator/SAMPLE.png')
    height, width, _ = template.shape
    # Create the output directory if it doesn't exist
    output_dir = 'GENERATED_CERTIFICATES'
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
    for index, name in enumerate(list_names):
        # Use a copy of the template to avoid modifying the original template
        temp_template = template.copy()
        # Use a simple font style
        font_style = cv2.FONT_HERSHEY_SIMPLEX
        font_size = 2
        text_thickness = 3
        text_position = (995, 761)  # Coordinates for name placement
        # Get the text width and height
        (text_width, text_height), _ = cv2.getTextSize(name, font_style, font_size, text_thickness)
        # Calculate the x-coordinate for center alignment
        x_coord = text_position[0] - text_width // 2
        # Write the name onto the template with center alignment
        cv2.putText(temp_template, name, (x_coord, text_position[1]), font_style, font_size, (0, 0, 0),
text_thickness, cv2.LINE_AA)
        # Save the certificate as a temporary JPG
        temp_jpg_path = f"temp_{name}.jpg"
        cv2.imwrite(temp_jpg_path, temp_template)
        # Convert the JPG to a PDF
        pdf_path = os.path.join(output_dir, f"{name}.pdf")
        pdf = FPDF(unit="pt", format=[width, height])
        pdf.add_page()
        pdf.image(temp_jpg_path, 0, 0, width, height)  # Use the same dimensions as the input template
        pdf.set_font("Arial", size=12)
        pdf.cell(200, 10, txt=name, ln=True, align='C')
        pdf.output(pdf_path, "F")
        # Remove the temporary JPG file
        os.remove(temp_jpg_path)
        # Save the certificate to the database
        certificate = Certificate(name=name, certificate_image=pdf_path)
        certificate.save()
        print(f'Processing certificate {index + 1}/{len(list_names)}')
    return HttpResponse('CERTIFICATES Generated Successfully')
```

3.2. Models

File: models.py

Python Copy code

```
from django.db import models
class Certificate(models.Model):
    name = models.CharField(max_length=255)
    certificate_image = models.ImageField(upload_to='certificates/')
    created_at = models.DateTimeField(auto_now_add=True)
```

3.3. URL Configuration in application

File: urls.py

Python Copy code

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.generate_certificates, name='generate_certificates'),
]
```

3.4. URL Configuration in main project

File: urls.py

Python Copy code

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('generator.urls')),
]
```

3.5. Running the Project

1. **Start the Django development server**:

   ```
   python manage.py runserver
   ```

2. **Access the certificate generation view**: Open a web browser and navigate to http://127.0.0.1:8000/.

Chapter 4

DETAILED EXPLANATATION

4.1. Reading the Excel Sheet

The project reads the names from an Excel sheet using Pandas. The file path is hardcoded, but it can be parameterized for more flexibility. The names are stored in a list for further processing.

4.2. Image Processing with OpenCV

OpenCV is used to read the template image and to overlay the names onto the certificate template. The coordinates for the text placement are predefined and can be adjusted as needed.

4.3. PDF Generation with FPDF

The modified template (with the name) is first saved as a temporary JPG file, which is then converted to a PDF using FPDF. This ensures that the certificates are saved in a widely accepted format.

4.4. Database Storage

The generated PDF paths are stored in a Django model for easy retrieval and management. This allows for maintaining a record of all generated certificates.

Chapter 5

ADVANTAGES AND DISADVANTAGES

5.1 Advantages :

1. Time Efficiency :

- Automates the certificate creation process, significantly reducing the time required to generate certificates manually.

2. Accuracy :

- Minimizes human errors by automating the data extraction and placement of names on certificates.

3. Consistency :

- Ensures that all certificates follow a uniform design and layout, maintaining a professional appearance.

4. Scalability :

- Capable of handling the generation of a large number of certificates, making it suitable for events with many participants.

5. Ease of Use:

- Provides a user-friendly interface, allowing users to generate certificates with minimal effort.

6. Record Keeping:

- Stores certificates in a database, making it easy to retrieve and manage records.

7. Flexibility:

- Allows customization of the certificate template, accommodating different designs and branding requirements.

8. Compatibility:

- Generates certificates in PDF format, ensuring they can be easily viewed, shared, and printed.

9. Cost Savings:

- Reduces the need for manual labor, leading to cost savings in the certificate creation process.

5.2 Disadvantages

1. Initial Setup Complexity:

- Requires initial setup and configuration, which may be complex for users without technical expertise.

2. Dependency on Software:

- Relies on various software components (Django, OpenCV, FPDF), which may require regular updates and maintenance.

3. Template Limitations:

- The predefined template design may not suit all needs, and significant customization may require additional effort.

4. Resource Intensive:

- Generating a large number of certificates, especially with high-resolution images, can be resource-intensive in terms of processing power and storage.

5. Potential for Bugs:

- Like any software application, there is potential for bugs and errors that could disrupt the certificate generation process.

6. Data Privacy Concerns:

- Handling personal data (names and potentially other details) requires ensuring data privacy and security measures are in place.

7. Learning Curve:

- Users may need to familiarize themselves with the system and its functionalities, which could involve a learning curve.

8. Internet Dependency:

- If the system is hosted online, it requires a stable internet connection for access and operation.

Chapter 6

FUTURE ENHANCEMENTS

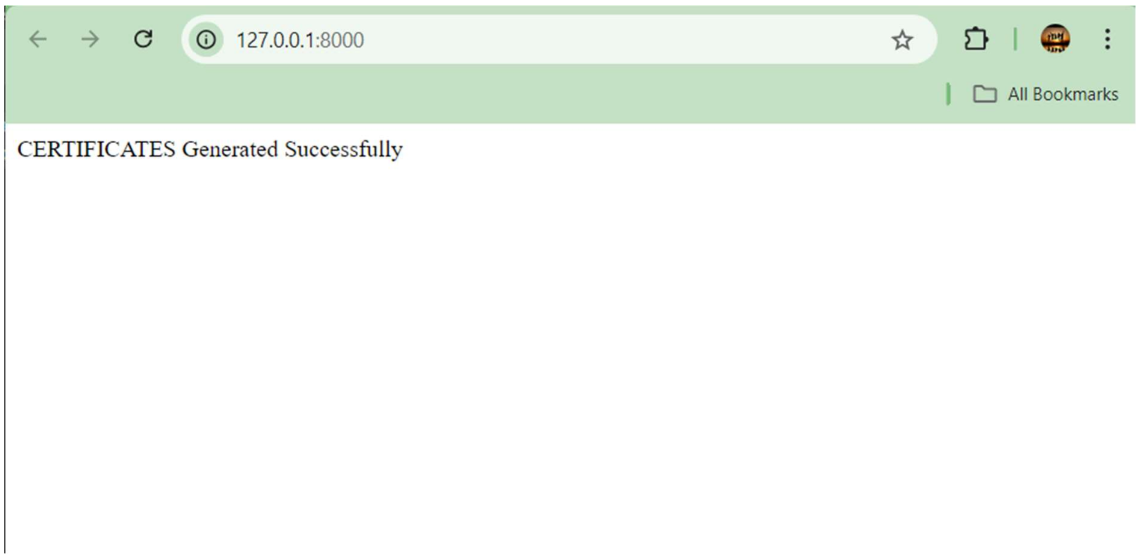
- **Template Management System**:
    - Implement a system for managing multiple certificate templates, allowing users to select and customize templates based on different events or requirements.
- **Batch Upload and Processing**:
    - Enable batch upload of Excel sheets and support concurrent processing to handle large datasets more efficiently.
- **Advanced Customization Options**:
    - Provide advanced customization options for text placement, font styles, colors, and adding additional elements like logos or QR codes to the certificates.
- **Integration with Third-Party Services**:
    - Integrate with third-party services for email delivery, cloud storage (like Google Drive, Dropbox), and printing services to automate the distribution and storage of certificates.
- **Mobile Application**:
    - Develop a mobile application to allow users to generate and manage certificates on the go.
- **Multilingual Support**:
    - Add support for multiple languages to cater to a broader audience and international events.
- **Enhanced Security Features**:
    - Implement enhanced security features, such as encryption of stored certificates, secure access control, and audit logs to track certificate generation activities.
- **Analytics and Reporting**:
    - Provide analytics and reporting features to give insights into certificate generation activities, such as the number of certificates generated, error rates, and user activity logs.
- **Integration with Learning Management Systems (LMS)**:
    - Integrate with popular LMS platforms like Moodle, Canvas, or Blackboard to automate certificate generation for completed courses and training programs.

Chapter 7

RESULTS

The Certificate Generator project successfully automates the process of creating and managing personalized certificates. By reading recipient names from an Excel sheet and overlaying them onto a pre-designed template using OpenCV, the system generates high-quality PDFs through FPDF. The certificates are stored in a database, ensuring efficient record-keeping and easy retrieval. This automation significantly reduces manual effort, minimizes errors, and ensures consistency across all certificates. The web interface, built with Django, provides a user-friendly way to initiate the process, making it accessible even for users with minimal technical expertise. The project demonstrates the integration of various technologies to deliver a scalable and efficient solution for certificate generation, suitable for educational institutions, training programs, and events. Future enhancements could further improve functionality, user experience, and system integration, making the tool even more versatile and powerful.

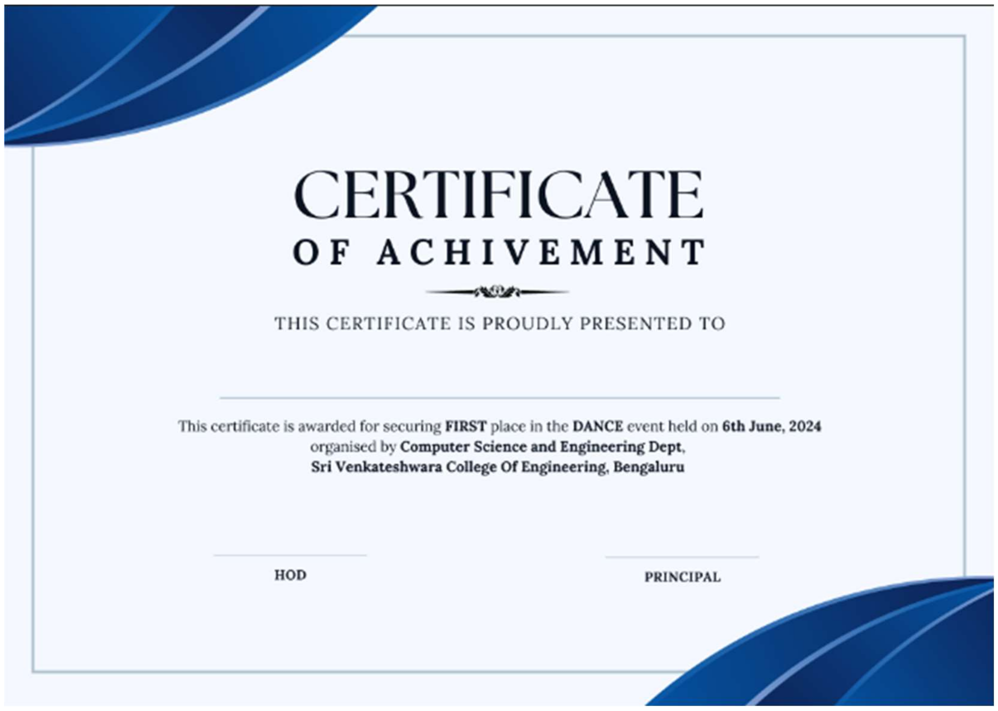
7.1 Output Screen :



CERTIFICATES Generated Successfully

7.3 Certificates

- Before



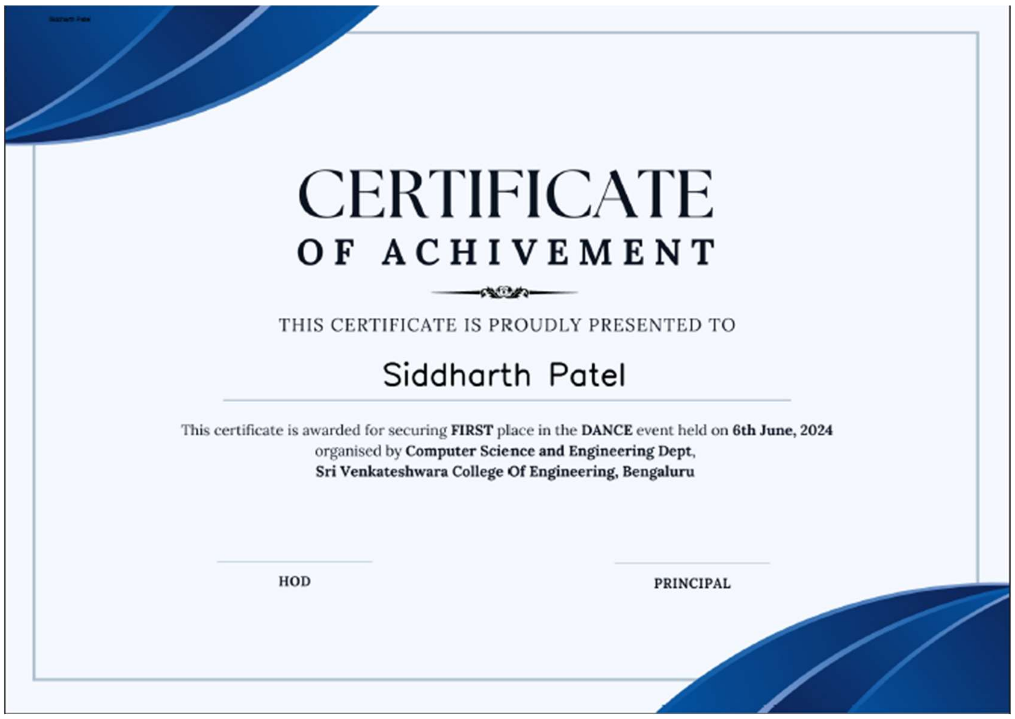
CERTIFICATE
OF ACHIVEMENT

THIS CERTIFICATE IS PROUDLY PRESENTED TO

This certificate is awarded for securing **FIRST** place in the **DANCE** event held on **6th June, 2024**
organised by **Computer Science and Engineering Dept,**
Sri Venkateshwara College Of Engineering, Bengaluru

HOD PRINCIPAL

- After



CERTIFICATE
OF ACHIVEMENT

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Siddharth Patel

This certificate is awarded for securing **FIRST** place in the **DANCE** event held on **6th June, 2024**
organised by **Computer Science and Engineering Dept,**
Sri Venkateshwara College Of Engineering, Bengaluru

HOD PRINCIPAL

Chapter 8

CONCLUSION

The Certificate Generator project exemplifies the power of automation in streamlining traditionally manual tasks, significantly enhancing efficiency and accuracy. By leveraging a combination of Django for web handling and database management, OpenCV for image processing, and FPDF for PDF generation, the project successfully automates the creation of personalized certificates. This integration of technologies results in a seamless workflow that reads recipient names from an Excel sheet, customizes certificate templates, and generates high-quality PDFs stored in a database.

One of the most significant advantages of this system is its ability to handle large volumes of certificates efficiently, making it ideal for educational institutions, training programs, and large-scale events. The automated process reduces the likelihood of human errors, ensures consistency across all certificates, and saves considerable time and effort compared to manual generation. Additionally, the user-friendly web interface ensures that even those with minimal technical skills can easily generate certificates.

The project's success highlights the importance of automation in administrative tasks and its potential to bring about significant improvements in operational efficiency. However, the initial setup and configuration require some technical expertise, and there is a dependency on various software components that may need regular updates and maintenance.

Future enhancements could further expand the system's capabilities, including a more sophisticated user interface, advanced customization options, batch processing, integration with third-party services, mobile applications, multilingual support, and enhanced security features. Implementing these improvements would make the Certificate Generator even more versatile and user-friendly, catering to a broader range of needs and scenarios.

In conclusion, the Certificate Generator project stands as a robust solution for automated certificate creation, offering significant benefits in terms of time savings, accuracy, and consistency. Its scalable and customizable nature ensures that it can be adapted to various requirements, making it a valuable tool for any organization needing to generate certificates efficiently.

REFERENCES

1. Django Documentation :- Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. For more details, visit the official Django documentation:[Djang Documentation] (https://docs.djangoproject.com/en/stable/).

2. OpenCV Documentation :- OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. Detailed documentation and tutorials can be found at: [OpenCV Documentation](https://docs.opencv.org/).

3.Pandas Documentation :- Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and data manipulation library or Python. For comprehensive information, refer to: [PandasDocumentation](https://pandas.pydata.org/pandas-docs/stable/).

4.FPDF Documentation :- FPDF is a free PHP class for generating PDF files. The Python implementation of FPDF can be found and detailed documentation is available at: [FPDF Documentation](http://www.fpdf.org/).

5. Python Imaging Library (Pillow) : - Pillow is a fork of the Python Imaging Library (PIL) and adds some user-friendlyfeatures. More information can be found at: [PillowDocumentation](https://pillow.readthedocs.io/en/stable/).

6. Certificate Template Design :- For best practices and guidelines in designing certificate templates, consult design resources and tools such as Canva or Adobe Spark: [Canva](https://www.canva.com/) | [Adobe Spark](https://spark.adobe.com/).

7. Excel File Handling in Python :- For insights and best practices on handling Excel files in Python using Pandas, refer to: [Pandas IO Tools](https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html).

8. Python Documentation :- The official Python documentation provides a comprehensive guide to Python's standard library and is an essential resource for developers: [Python Documentation](https://docs.python.org/3/).

9. Django File Uploads :- For handling file uploads in Django, including ImageField, refer to the relevant section of the Django documentation: [Django File Uploads](https://docs.djangoproject.com/en/stable/topics/files/).

10. Web Development Best Practices :- For best practices in web development and ensuring security, performance, and scalability, refer to resources like MDN Web Docs: [MDN Web Docs](https://developer.mozilla.org/en-US/).

These references provide a solid foundation for understanding the technologies and methodologies used in the Certificate Generator project and offer additional resources for further exploration and development.