MSc WORKFLOW AND DOCUMENT MANAGEMENT SYSTEM BUILT WITH PYTHON (DJANGO)

Department of Electronics and Computer Engineering

Pulchowk Campus, Lalitpur

Date: August 31, 2023

The Quadrupole

1. INTRODUCTION:

Welcome to the documentation for the MSc Workflow and Document Management System, a cutting-edge solution developed for the Institute of Engineering (IOE), Pulchowk Campus, specifically catering to the needs of the Department of Electronics and Computer Engineering. This project, which was initiated by our predecessors, has now been enhanced and evolved through the collaborative efforts of our team under the guidance of Dr. Aman Shakya, our esteemed instructor for the Software Engineering course.

Project Overview:

The MSc Workflow and Documentation System serves as a pivotal tool for the management and automation of workflows, along with document organization, within the academic sphere. Originally conceived and utilized within local environments by department coordinators, the system has now been expanded to accommodate multi-user functionality and has been successfully deployed for practical use.

Purpose and Objectives:

The primary objective of this system is to streamline and optimize the processes integral to the Master's program at IOE, Pulchowk Campus. By seamlessly integrating technology, we aim to revolutionize the way academic workflows are managed and documented. From submission and

review cycles to administrative approvals, the MSc Workflow and Documentation System offers an efficient and user-friendly platform for coordinating these intricate tasks.

Technological Framework:

The project has been meticulously crafted using a modern and robust technological stack. On the backend, we employ the renowned Django framework, leveraging the power of Python to create a reliable and extensible foundation. The frontend is seamlessly realized through Django's versatile admin interface, combining JavaScript, CSS, and HTML to deliver an intuitive user experience. The system's data management relies on the efficiency of the SQLite database.

Evolution and Enhancement:

Originally confined to local machine usage, our project's mandate was to elevate it into a comprehensive and deployable solution. Through rigorous development, testing, and collaboration, we have transformed the project from a single-user tool to a dynamic multi-user system, accessible remotely. This transformation reflects our dedication to enhancing academic processes and promoting digitalization within our educational institution.

As you delve into this documentation, you will gain insights into the system's architecture, installation procedures, usage guidelines, customization options, and more. Whether you are an academic staff member, a student, or a technology enthusiast, we invite you to explore the intricacies of the MSc Workflow and Documentation System and join us in embracing the future of academic management.

2. GETTING STARTED:

This section will guide you through the process of setting up and running the MSc Workflow and Documentation System on your local machine. By following these steps, you'll be able to explore the project's functionalities firsthand.

Installation Instructions:

Before you begin, make sure you have Python and Django installed. You'll also need the project's dependencies listed in the requirements.txt file. Run the following commands in your terminal:

```
 Install Python (if not already installed)
# Visit https://www.python.org/downloads/ for installation instructions

# Install Django (if not already installed)
pip install Django
```

Cloning the Repository:

To get started, clone the project repository using the following commands:

```
# Clone the repository
git clone https://github.com/darpankattel/msc-workflow-document-ms.git

# Navigate to the project directory
cd msc-workflow-document-ms
```

Configuration Steps:

No additional configuration steps are required for this project. It's designed to work seamlessly without any complex setup.

Running the Project Locally:

To run the project locally, follow these steps:

    a. Install Virtual Environment:

If you haven't installed the venv module, use the following command:

```
pip install virtualenv
```

b. Create and Activate Virtual Environment:

Create a virtual environment named "env" and activate it:

```
python3.6 -m venv env
# ORRRRRR
Python –m venv env –p "your python path to python 3.6.5"
source env/bin/activate   # On Windows: env\Scripts\activate
```

Install Requirements:

Install the project dependencies using the requirements.txt file:

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

Run the Server:

Start the Django development server:

```
python manage.py runserver
```

Access the Project:

Open your web browser and navigate to http://127.0.0.1:8000/ to access the MSc Workflow and Documentation System.

Note: Ensure that you use **Python version 3.6.5** as the interpreter when creating the virtual environment. Other versions might lead to compatibility issues.

By following these steps, you should now have the MSc Workflow and Documentation System up and running on your local machine. Feel free to explore its features and functionalities as you embark on this journey of academic process automation.

If you encounter any issues or have questions, please refer to the troubleshooting section or reach out to our support team.

Happy exploring!

USAGE:

Welcome to the usage guide for the MSc Workflow and Documentation System. This section will help you navigate the essential steps to interact with the system effectively.

1. Login to the Admin Dashboard: To access the admin dashboard, follow these steps:

- **Create Superuser:** Open your terminal and navigate to the project directory. Run the following command to create a superuser:

  python manage.py createsuperuser

- **Login as Superuser:** Launch the development server using **python manage.py runserver**, and then open your web browser and go to **http://127.0.0.1:8000/admin/**. Log in using the following credentials:
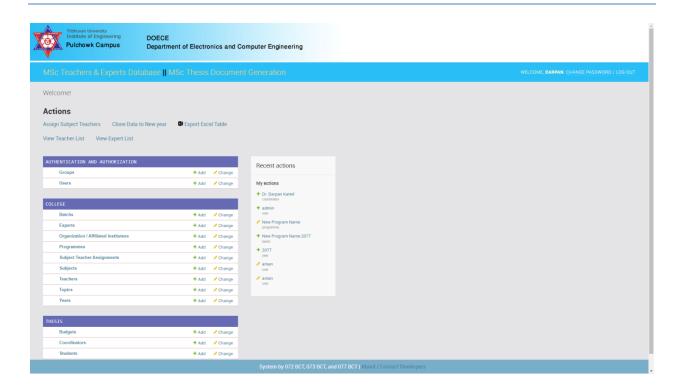
  - Username: admin

  - Password: pulchowk

2. Configure Your Account: After logging in as the superuser, you can configure your own user account as a coordinator:

- Go to the "Users" section in the admin dashboard.

- Locate and edit your user account.

- Assign the "Coordinator" role to your account.

By making your user account a coordinator, you'll be able to access and utilize the various features and functionalities available in the MSc Workflow and Documentation System.
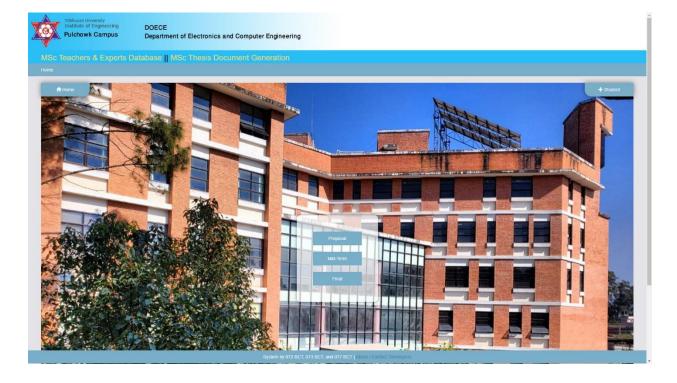
3. Use:

a.   Home Interface:

It contains the links for database control and thesis documents generation. Also, the database can also be controlled through this interface.

b.   Thesis Document Generation Interface:

It contains three link with one add student button; proposal, mid term and final.

3. CUSTOMIZATION:

Customization of MSc Workflow and Documentation System

The MSc Workflow and Documentation System is designed with flexibility in mind, allowing you to easily tailor its components to your specific requirements. Here's how you can customize various aspects of the project:

Customizing the Django Admin Dashboard:

The Django admin dashboard can be customized to match your institution's branding and user experience. To achieve this:

• **Static Files:** We Customized the look and feel by adding our own CSS and JavaScript files. These was done by freezing the static files:

```
python manage.py collectstatic
```

• **HTML Templates:** Create new HTML templates or modify existing ones as needed. These templates are stored in the templates directory and can be adjusted to match your desired layout and design.

Modifying Existing Workflows or Adding New Ones:

To modify existing workflows or add new ones, follow these steps:

1. **Views (views.py):** Update or create views that define the logic and behavior of your workflows. These views handle data processing, interactions, and render templates.

2. **URLs (urls.py):** Define URL patterns that map to your views. These patterns determine how users access the various workflows in your system.

3. **Models (models.py):** Modify the database models to accommodate any changes or additions required for your workflows. Ensure that the data structures align with your new features.

4. **Admin (admin.py):** This is a crucial step when modifying workflows. Customize the admin interface to provide an intuitive and user-friendly experience for managing workflows and associated data.

Customizing Database Configurations:

To customize the database configuration, you'll work with the settings.py file. Django makes it straightforward to adapt the database to your specific needs:

1.   **Database Engine:** In the settings.py file, you can configure the database engine you wish to use. By default, your project uses SQLite, but you can switch to other database systems like PostgreSQL or MySQL by adjusting the DATABASES setting.

2.   **Database Name and Credentials:** Modify the database name, username, password, and other credentials as required by your chosen database system.

3.   **Database Migrations:** After making changes to your models or database configuration, create and apply database migrations using:

```
python manage.py makemigrations

python manage.py migrate
```

Customizing the MSc Workflow and Documentation System allows you to tailor it to the specific needs of your institution, enabling seamless integration with your existing workflows and processes.

Remember, while customization provides immense flexibility, it's important to maintain consistency and follow best practices to ensure the integrity of the system.

4. CONTRIBUTING:

We greatly appreciate your interest in contributing to the MSc Workflow and Documentation System. Your contributions can help improve the project and make a positive impact on the academic community. Here's how you can get involved:

Pulling and Pushing Changes:

Fork the Repository:

Start by forking the MSc Workflow and Documentation System repository to your own GitHub account. This will create a copy of the repository under your account.

Clone Your Fork:

Clone your forked repository to your local machine using the following command:

```
git clone https://github.com/your-username/msc-workflow-document-ms.git

cd msc-workflow-document-ms
```

Pull Changes from Upstream:

Before you start making changes, it's a good practice to synchronize your repository with the original repository. This ensures you have the latest updates:

```
git remote add upstream https://github.com/darpankattel/msc-workflow-document-ms.git

git fetch upstream

git merge upstream/main
```

Make Your Contributions:

Create a new branch for your changes:

```
git checkout -b your-branch-name
```

Make your desired changes to the project.

Push Your Changes:

Once you're done with your changes, push them to your forked repository:

```
git push origin your-branch-name
```

Create a Pull Request:

Go to your fork on GitHub and click on "New Pull Request". Describe your changes and submit the pull request. We'll review your contributions and merge them if they align with the project's goals.

Important Note:

While we encourage contributions, please keep in mind that we may not be able to provide ongoing support for the project. Considering this, it's recommended that contributors create their own repositories for their work. This ensures that you have control over your contributions and can continue maintaining them independently if necessary.

Thank you for considering contributing to the MSc Workflow and Documentation System. Your efforts can make a valuable difference in advancing academic workflow automation.

5. <u>DATABASE SCHEMA:</u>

<u>Thesis</u>

Fig.: Thesis App Entity Relationship

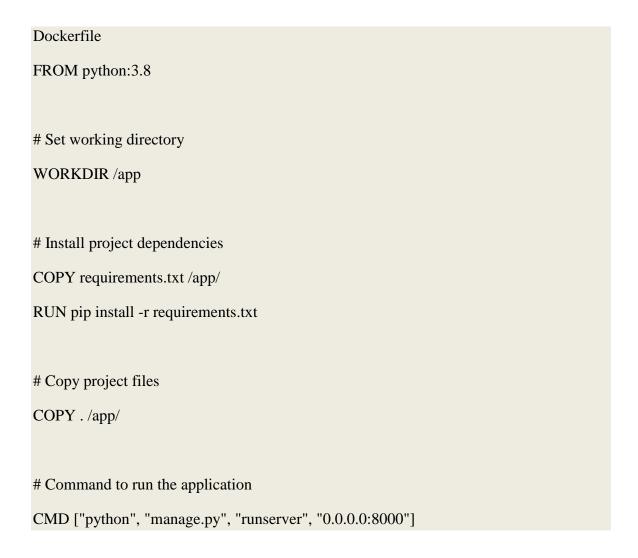Fig.: College App Entity Relationship

6. DEPLOYMENT:

The "M.Sc Workflow and Document Management System" has been successfully deployed using Docker, a platform that enables us to package our application along with its dependencies and configurations into a single container. This deployment method offers portability, consistency, and ease of management, making it an ideal choice for our project.

Dockerization

To deploy our project using Docker, we followed these steps:

1.**Dockerfile Creation:** We created a Dockerfile in the project directory to define how our application should be built within a Docker container. This file includes instructions to install necessary dependencies, copy project files, and set up the environment.

```
Dockerfile

FROM python:3.8


# Set working directory

WORKDIR /app


# Install project dependencies

COPY requirements.txt /app/

RUN pip install -r requirements.txt


# Copy project files

COPY . /app/


# Command to run the application

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

2.**Building the Docker Image:** With the Dockerfile in place, we used the docker build command to create a Docker image for our application. This image contains everything needed to run our project, including Python dependencies, the Django application, and the associated configurations.

```
docker build -t msc-workflow-app .
```

3.Running the Docker Container: Once the Docker image was built, we used the docker run command to start a Docker container based on the image. We exposed port 8000 from the container to the host machine so that we can access the application through a web browser.

```
docker run -d -p 8000:8000 msc-workflow-app
```

Accessing the Application

After deploying the project using Docker, the "M.Sc Workflow and Document Management System" can be accessed by opening a web browser and navigating to http://localhost:8000. This URL points to the running Docker container and allows users to interact with the application seamlessly.

Benefits of Docker Deployment

Deploying our project using Docker offers several advantages:

- Isolation: Docker containers encapsulate the application and its dependencies, ensuring that it runs consistently across different environments.
- Portability: Docker images can be easily shared and moved between different development, testing, and production environments.
- Scalability: Docker containers can be scaled up or down based on demand, making it easier to manage varying levels of traffic.
- Version Control: Docker images can be versioned, allowing us to roll back to previous versions if needed.
- Consistency: Docker eliminates the "it works on my machine" problem by providing a consistent environment for development and deployment.

By leveraging Docker, we have streamlined the deployment process of our "M.Sc Workflow and Document Management System," making it more efficient and reliable.

7. <u>KNOWN ERRORS:</u>

a. Tangled Login Page (At Front-End)

Explanation:

The login page of the website appears tangled, with its footer positioned incorrectly. Additionally, the login page, while simple, lacks a standard appearance. The footer placement issue adds to the disarray, creating an impression of disorderliness that contrasts with the intended simplicity of the design.

b. Finals Thesis Defence List (At Front-End)

Url: /docgen/final_thesis_defence_list

I

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING, PULCHOWK CAMPUS
Department of Electronics and Computer Engineering
New Program Name

मिति:- 14 th magh 2080

श्रीमान् क्याम्पस प्रमुख ज्यू,
पुल्चोक क्याम्पस, इ॰ अ॰ सं॰,
पुल्चोक, ललितपुर ।

विषय: - Final Thesis Defense को Thesis Title हरु बारे ।

उपरोक्त सम्बन्धमा यस त्रि॰वि॰ इन्जिनियरीङ अध्ययन संस्थान, केन्द्रीय क्याम्पस पुल्चोक अन्तर्गत इलेक्ट्रोनिक्स तथा कम्प्युटर इन्जिनियरीङ विभागमा संचालित New Program Name कार्यक्रम अन्तर्गत दोस्रो वर्ष दोस्रो खण्डको परीक्षा मध्ये 2080 ब्याचका विद्यार्थी जम्मा 0 जनाको मिति 15th magh 2080 गते सम्पन्न हुने Final Thesis Defense प्रयोगात्मक परिक्षाको Thesis Title तपसिल बमोजिमको रहेको जानकारी गराउँदछु ।

S. N.	Title of Thesis	Name and Roll No. of Students	Name of the Supervisor
)		0	

Dr. Darpan Kattel
प्रोग्राम कोअर्डिनेटर

बोधार्थ :-
१) इलेक्ट्रोनिक्स तथा कम्प्युटर इन्जिनियरीङ विभाग
२) आर्थिक शाखा, पुल्चोक क्याम्पस, इ॰ अ॰ सं॰

Explanation:

As we can see here, the data entered in the site is not exported to the file. This is definitely a bug in the back-end code.

ACKNOWLEDGEMENT:

From

Team Quadrupole (PUL 077 BCT)

Darpan Kattel (077 BCT 099) 077bct099.darpan@pcampus.edu.np

Bishnu Datt Badu (077 BCT 098) 077bct098.bishnu@pcampus.edu.np

Javed Ansari (077 BCT 033) 077bct033.javed@pcampus.edu.np

Bibek Sunar (077 BCT 097) 077bct097.bibek@pcampus.edu.np

Last deployed at: https://mscwfadms.coderslap.com/

Github Link: https://github.com/darpankattel/msc-workflow-document-ms