

PU ONLINE EXAMINATION FORM REGISTRATION

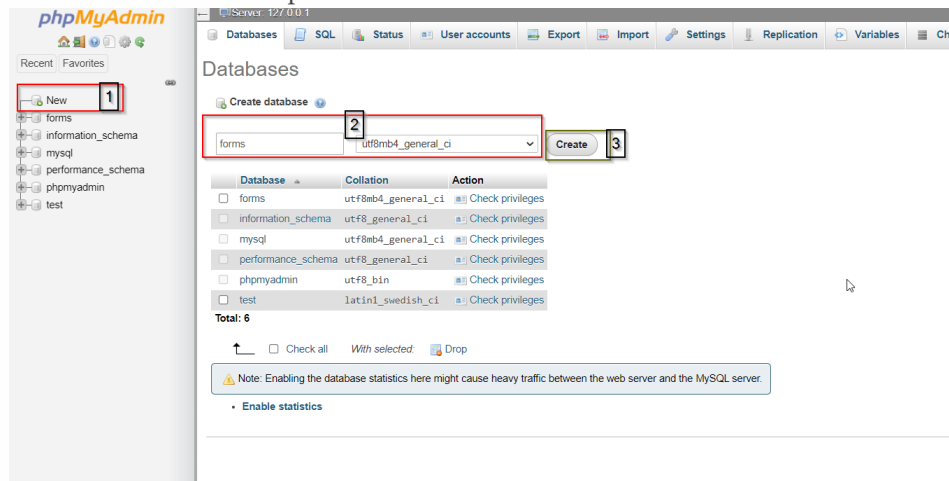
Installation guide

• STEP1: Installing DBMS

1. For Windows: Install Xampp and your are ready to go.

• STEP2: Create A Database 'forms'

1. For xampp, go to localhost/phpmyadmin Then click on New, then create a database called 'forms' as shown in picture below.



• STEP3: SETUP PYTHON AND PIP

FOR WINDOWS

1. Install latest python from <https://www.python.org/>
 2. After installation, make sure `pip` is installed. You can execute `pip --version` on terminal to verify this.
- If not installed, download this file [get-pip.py](#). then you can execute `python get-pip.py` to install pip.

• STEP4: SETUP PIPENV SHELL

1. Go to the folder where the project is situated.
2. Openup a terminal there, for Windows:
`Shift + Right-Click and click on powershell window`
3. Type `pipenv shell` to create a pipenv shell

This creates a isolated environment for your python projects so that the project specific modules would not collide with system-wide installed python modules

4. Make sure you are in the right directory (i.e. be in the directory where project files are)

• STEP5: INSTALL NECESSARY MODULES

1. Following the Step 4, now type: `pipenv install -r requirements.txt` to install all necessary modules at once.

• **STEP6: SETUP DATABASE CONNECTION**

1. Now, open xampp and start mysql and apache service.

2. Then open `config.py` file from project and edit the line

```
`SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:@localhost:3306/forms'
```

3. Replace `root:` with your database credentials in the format `username:password`

4. Replace `localhost:3306` with your database connection url. It is most likely to be the same.

5. The resultant line should be :

```
SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://username:password@url_for_db:port_value/form
```

6. Save the file

• **STEP7: SETUP DATABASE**

1. Now we will create database tables and populate the tables with necessary data

2. The credentials of admin will be as follows:

Admin credentials:

email: `admin@email.com`

password: `admin`

Note : Donot worry, we donot save plain password. We use bcrypt to generate encrypted passwords.

3. Execute, on the powershell(command line): This will create tables and populate tables

```
python database_setup.py
```

• **STEP8: RUN THE WEB APPLICATION**


1. Now, on the command line, execute:

```
set FLASK_APP=app.py
```

2. Then, execute:

```
flask run
```

3. Now, the web app should run at <http://localhost:5000> and you will see a similar screen

A screenshot of a terminal window titled 'MINGW64: c:/Users/Roshan/Desktop/Project-I'. The prompt is 'Roshan@DESKTOP-8REG71U MINGW64 ~/Desktop/Project-I (master)'. The user enters '\$ set FLASK_ENV=app.py'. The prompt changes to 'Roshan@DESKTOP-8REG71U MINGW64 ~/Desktop/Project-I (master)'. The user enters '\$ flask run'. The output shows: '* Environment: production', 'WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.', '* Debug mode: off', and '* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)'. A cursor is visible at the end of the last line.

```
MINGW64: c:/Users/Roshan/Desktop/Project-I
Roshan@DESKTOP-8REG71U MINGW64 ~/Desktop/Project-I (master)
$ set FLASK_ENV=app.py
Roshan@DESKTOP-8REG71U MINGW64 ~/Desktop/Project-I (master)
$ flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```