EE468 – Team A

Design Software User Guide

For use with Django Database Application

Document 800c0d4b

Revision A: April 2022

Django Application - Contents

### Contents

1. About This Document

[Intended audience 3](#_Toc102122935)

[Additional Django Application documents 3](#_Toc102122936)

[System requirements 3](#_Toc102122937)

[2. Overview of the Software 4](#_Toc102122938)

[The Django Database Application Software 4](#_Toc102122939)

[3. Using the Django Database Software 5](#_Toc102122940)

[Download required software 5](#_Toc102122941)

[Sign into your account 5](#_Toc102122942)

[Select desired query 5](#_Toc102122943)

[Admin Page 6](#_Toc102122944)

[Professor Page 6](#_Toc102122945)

[Student page 7](#_Toc102122946)

Django Application - 1. About this document

1. About this document

# Intended audience

This document is intended to be used by system administrators and others experienced in working with database applications. This document will be used in concert with Django open-source web framework software to configure your organization’s database application. The configuration software is available for download at https://github.com/lukaszmp/EE468-Project.

# Additional Django Application documents

* For general instructions for using and maintaining the Django Database Application see the User Design Manual
* For information about implementing the server application, see Communication Protocols.

All of these documents are available at <https://github.com/lukaszmp/EE468-Project>

# System requirements

* Wireless network and server application
* Computer with the following installed on it:
* Django open-source web framework
* Newest version of Python
* MySQL
* Access to MySQL University Database

**NOTE:**

D

Django Application - 2. Overview

# 2. Overview of the Software

# The Django Database Application Software

The Django Database Application software is a tool for creating a web-based University Course Management System. This software will make use of the pre-designed SQL based university database our team designed early in the school semester. The system will utilize the Django web framework to connect to MySQL and hold together all the components including loading data from the database and dynamically creating a webpage to display the data. The user interface will be built using Django templates.

The software will separate users based on their role in the university database and give permissions to view different data reports based on this role. These roles are separated into three separate categories: Admin, Professors, and Students respectively.

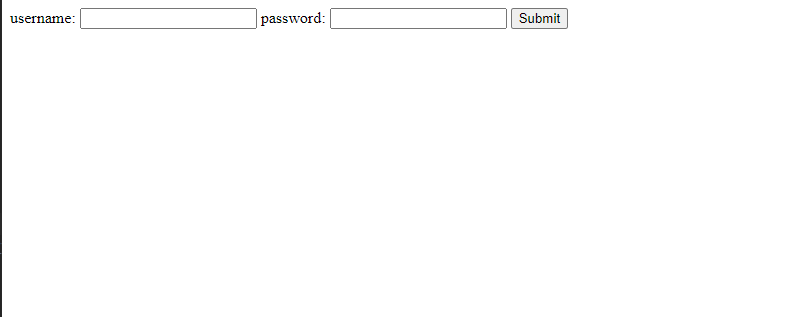
# 3. Using the Django Database Software

# Download required software

If you have not already, download the necessary software including Django, MySQL, the University Database made within MySQL, and the GitHub project containing the software that can be found at: https://github.com/lukaszmp/EE468-Project

# Sign into your account

1. Upon launching the software, the user should go to their search engine of choice and enter the link: <http://127.0.0.1:8000/usertype> to reach the login page.
2. Upon entering the link into the user’s search bar, the user will be greeted by a “login screen” which will prompt the user for a specific user name and password that can be confirmed by hitting Submit.
3. The user must be careful in entering their login information, as the use of a correct username with an incorrect password 5 or more times will result in an account lockdown for security purposes. This lockdown can then be removed through the correct user’s linked email account (may not be functional at time of submission).
4. After a successful login the system will check the corresponding permissions of the account and send the user to the appropriate page, being either Admin, Professor, or Student.
5. The user will be able to return to the login screen by clicking the escape button on their keyboard



**Figure 1-1 Django Database Software Sign in Page**

**Important Notes:**

**settings.py and views.py references to passwords may need to be changed when utilizing the project on a new computer.**

**Default Usernames for website: student, professor, admin Respectively**

**Default password for all users on website: EE468CS460**

**Default password for the Database is : password12**

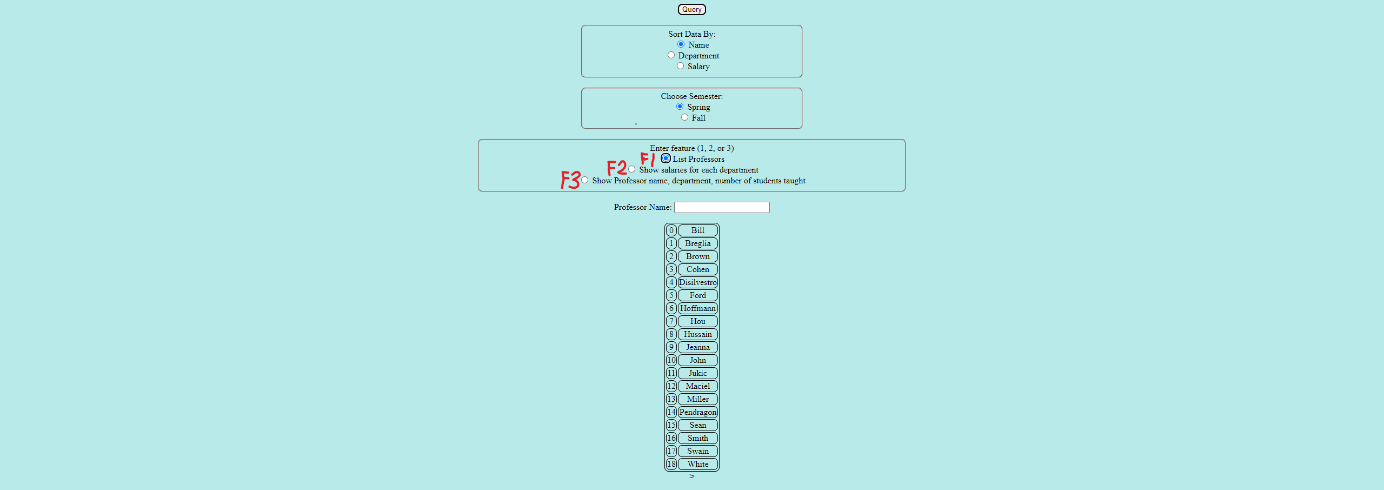
**(this password will likely need to be changed)**

## Select desired query

1. After the user is directed to their intended page, they will be greeted by several query options, each option having their own corresponding entry fields and submission buttons as needed
2. If the user enters valid variables corresponding to any given query and hits the submit button, the software will return a visual table containing the desired results for the user to viewing

## Admin Page

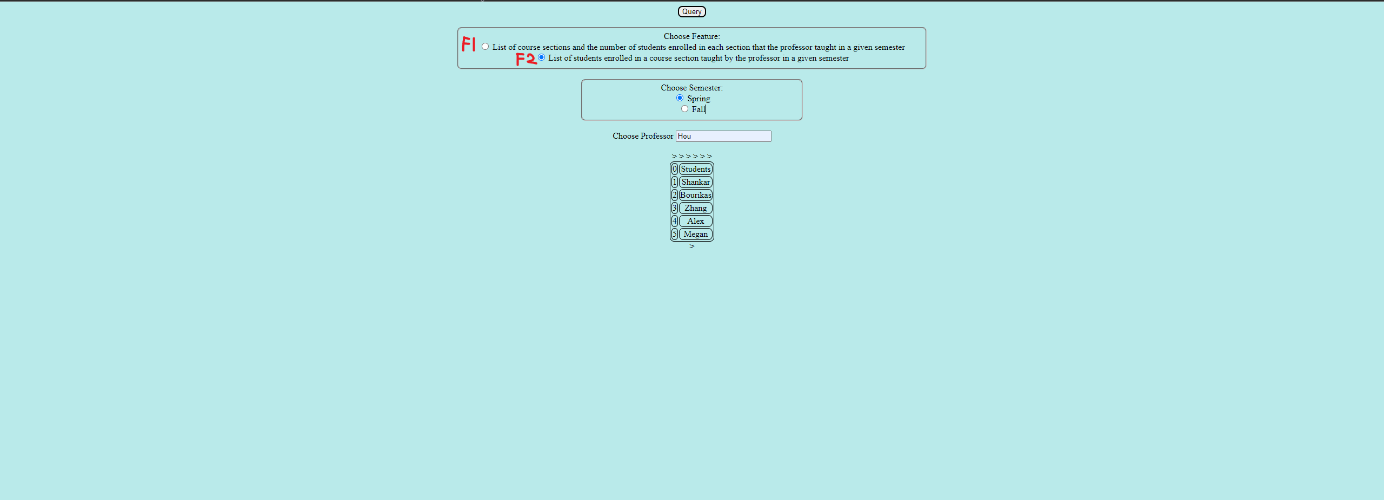
1. Upon gaining access to the admin page the user is presented with three pre-set functions to select from, they must then decide which function they would like to use, enter valid information in corresponding fields and click the submit button (page can be found at <http://127.0.0.1:8000/usertype/admin> as well)
2. Function 1 (represented by F1 in figure 1-2) will create a list of professors sorted by one of the following criteria selected by the user with a radio button: 1. By name, 2. By department, 3. By salary
3. Function 2 (represented by F2 in figure 1-2) will create a table of min/max/average salaries by department specified by the user in the given text field
4. Function 3 (represented by F3 in figure 1-2) will create a table of professor names, departments, and total number of students being taught by the respective professor within a user selected semester represented by a radio button of either spring or Fall semesters



**Figure 1-2. Django Database Software Admin Page**

## Professor Page

1. Upon gaining access to the professor page the user is presented with two pre-set functions to select from, they must then decide which function they would like to use, enter valid information in corresponding fields and click the submit button (page can be found at <http://127.0.0.1:8000/usertype/professor> as well
2. Function 1 (represented by F1 in figure 1-3) will create a table of course sections and the number of students in each section taught by the given professor within a given semester. The user would have to select a radio button corresponding to the corresponding semester before submitting.
3. Function 2 (represented by F2 in figure 1-3) will create a table of student’s names that are enrolled in a selected course section within a selected semester that is taught by the given professor. The user would select the desired semester with a given radio button.



**Figure 1-3. Django Database Software Professor Page**

## Student page

1. Upon gaining access to the student page the user is presented with one pre-set function, they must then enter valid information in corresponding fields and click the submit button (page can be found at <http://127.0.0.1:8000/usertype/student> as well)
2. Function 1 (the only function displayed in figure 1-4) will create a table of available course sections offered by a department within a given year and semester. The user would need to select the desired department, year, and semester before clicking the submit button.



**Figure 1-4. Django Database Software Student Page**