

# Feedback Generation Form

Gade Sai Shravan



APRIL 27, 2023

BACHELOR OF SCIENCE (HONORS) IN COMPUTER SCIENCE  
SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING,  
MUDDENAHALLI CAMPUS

# ***Tables of contents***

Acknowledgment .....	3
Problem Domain.....	4
Introduction .....	5
Framework used.....	6-7
Libraries.....	8
About and Process Flow.....	9-10
Overview.....	11-22
User Registration.....	11
Login.....	12-13
Feedback /Regdno.....	13-19
Submit / Print.....	20
Uploading Data.....	21-22
Django Admin.....	23-24
Approach .....	25-28
Conclusion.....	29
Bibliography.....	30

# **Acknowledgment**

we would like to extend our sincere thanks to Sir.B.Venkata Ramana sir for his supportive, thoughtful and caring guidance, for this project would not be what it is without him.

we would like to extend our gratitude to Bhagawan Sri Sathya Sai Baba for giving us the opportunity to study Computer Science, for we would not have any knowledge without him, for development of this project.

# **Problem Domain :**

Problem domain of this project is to build a web application which helps the teachers to give the feedback to the students and issue the feedback forms to the students by printing them. The application should also has the feature to upload data of the students details and fetch the students data from the database when giving the feedback.

## **SRS :**

**Purpose :** Purpose of the product is to help the teachers to give feedback students easily.

**Requirements :** Skills in django frontend framework(python),Html,CSS,bootstrap.

# Introduction

We group of three were assigned with the task of creating a web application that allows the teachers to upload the data of the students which contains the marks of the students for their current semester and save them to the database, and helps the teachers to give the feedback on students easily.

Moving forward, the second key of our project is, when the teacher uploads the students data to the database, the data gets saved in the particular table of the database. Now, when teacher wants to give feedback to the student, teacher selects the particular semester, then teacher is asked to provide the Regdno of the student and when he teacher clicks submit, the details of the student are fetched from the database and two textboxes are available to give feedback. Then we provide two options (submit and print), when clicked on print the form can be printed, and when clicked submit, the feedback which given by teacher gets saved at that particular student Regdno.

# **Framework used :**

The framework used for our project is Django.Django is a high-level python web framework that encourages rapid development and clean, pragmatic design.It takes care of much of the hassle of web development.

**Functionality :** A Django project is a collection of different applications having different functionalities.

Django allows us to map a requested URL from a user to the code that's actually meant to handle it and it also allows us to create that requested HTML dynamically.

**VIEWS:** Django views are Python functions that takes http requests and returns http response, like HTML documents.Views are usually put in a file called **views.py** located on your app's folder.

**MODELS:** In Django, data is created in objects, called Models, and is actually tables in a database.Models are stored in **models.py** file.

When we create a model in django with certain fields the database must know the fields to be added to the database table for that we run the,

**py manage.py makemigrations** command, then the django creates the table in the database of that class name in file. To add the fields in the table that are in class we run the following command, **py manage.py migrate**, then the fields in the class mentioned will be added to the fields in the table of the database.

Django has its inbuilt database in which the user can work easily and develop much of the web applications.

**Django Admin:** Django Admin is a really great tool in Django, it is actually a CRUD\* user interface of all your models! It is free and comes ready-to-use with django. It is like a admin where a superuser is created and handles everything in the project happening and checks whether everything is under his control.

**Django Forms:** Django comes with the inbuilt feature of creating forms and helps the user to create multiple forms in one single project. One major advantage that django forms gives us is that Validation. Itself checks the fields entered are correct or wrong and if wrong, it does not allow the user to access the application.

## ***Database :***

Django comes with an inbuilt database layer that provides an object-relational mapper (ORM) to interact with various databases. By default, Django uses SQLite as its database backend, which is a lightweight and easy-to-use database engine that stores data in a single file.

Django's ORM allows you to define your database schema using Python classes and methods, which makes it easy to create and modify database tables and relationships. You can also use Django's ORM to perform CRUD (Create, Read, Update, and Delete) operations on your data, and the ORM handles the database operations in a safe and efficient way.

Overall, Django's inbuilt database provides a powerful and flexible way to store and retrieve data in your web applications, and the ORM layer makes it easy to work with databases without having to write SQL code directly.

## **Libraries Used :**

We have used two libraries in the development of this web application, they are :

1. Pandas.
2. Import and Export.

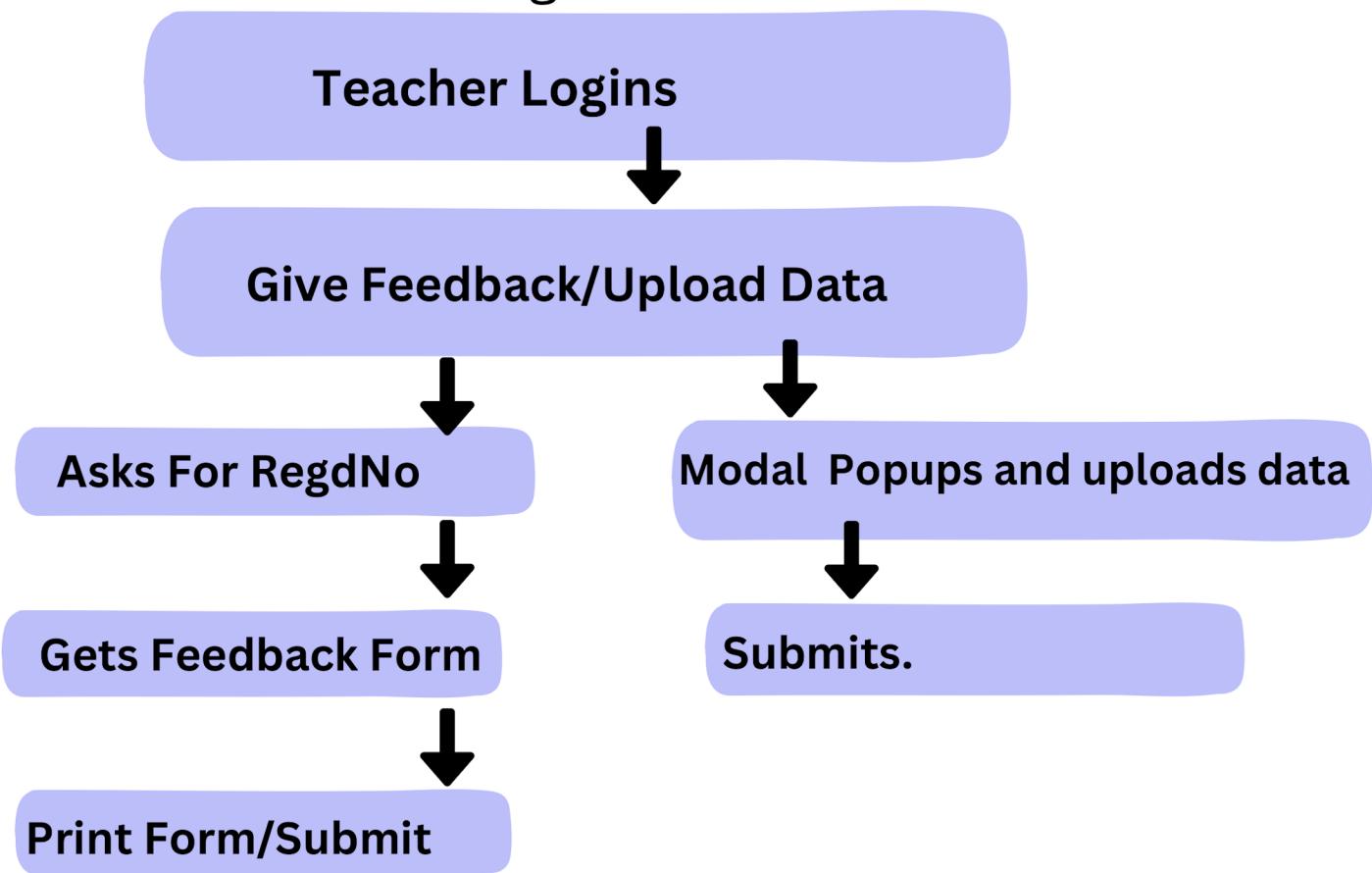
**Pandas** : Pandas is a popular Python library used for data manipulation and analysis. It provides data structures and functions for working with structured data, such as spreadsheets, CSV files, and SQL tables. In the context of uploading data in an application, pandas can be used to read and process data from a file, validate the data, and save it to a database or data store. The teacher uploads the excel file containing student marks in application through pandas.

**Import and Export** : In Django's admin context, import and export functionality can be implemented using the django-import-export library. This library provides a simple and customizable way to import and export data from Django models to various file formats like CSV, Excel, JSON, etc. Django superuser has the this feature where he can import or export the files in django and store them at particular model.

## ***About and Process Flow :***

The introduction above gave is an overview of what the project does. In this section we clearly see how the project works in the form of flow-diagrams:

*Figure 1*



The above mentioned work flow gives the overall view of the web application. Firstly, the user gets registered to an application. Then the user is asked to get login into the application to give feedback, or to upload the data.

When user is logged in, he has two options one is for to give feedback and the other is to upload the data.

when user selects the class for giving feedback it asks the regdno to enter, when user enters the regdno and clicks the submit button then the details of the student is fetched from the data base to the feedback form.

Then the whole form is created with two empty textboxes for the teachers to give the feedback on the students.

When the teacher gives the feedback for the students the feedback alone given is saved in the database at that particular student.

Then the teacher has the option to print the overall feedback form with all necessary information and issue the feedback to the students.

**OverView :**

# User Registration :

## Register

Username\*  
 Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email\*

Password\*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation\*  
 Enter the same password as before, for verification.

If you already have an account, please [login](#) here.

Django comes with a pre-built register form called User Creation Form that connects to the pre-built model User. However, the User Creation Form only requires a username and password (password1 is the initial password and password2 is the password confirmation).

Django-registration is an extensible application providing user registration functionality for Django-powered Web sites.

We use `is_valid()` when required to validate complete form-data. This validation will check for Python-datatypes. This function will return True or False (Python Data Types) in return.

## User Login :



Django comes with an inbuilt authentication system. The Django authentication system handles both authentication and authorization. Briefly, authentication verifies a user is who they claim to be, and authorization determines what an authenticated user is allowed to do. Here the term authentication is used to refer to both tasks.

The auth system consists of:

- Users
- Permissions: Binary (yes/no) flags designating whether a user may perform a certain task.
- Groups: A generic way of applying labels and permissions to more than one user.
- A configurable password hashing system
- Forms and view tools for logging in users, or restricting content.

When the user tries to log in, the django authentication system checks whether the user is registered or not. If yes, the user can log in.

User gets an option to get registered to an application. When user gets registered, the username,password, mail and other credentials will be saved in the django users database. When user tries to login, the django '**auth**' process the username and password enters, if the credentials are valid then the user gets logged in to the application.

## ***Uploading data/giving feedback :***

### Sri Sathya Sai Institute of Higher Learning

welcome to Teacher's Portal

welcome, bvr!

[Logout](#)

Please select the classes to give feedback for the students :

UG:

[I Bsc](#) [I BBA](#)

[II Bsc](#) [II BBA](#)

[III Bsc](#) [III BBA](#)

PG:

[I MSc](#) [II MSc](#)

Please select the classes to upload the data :

UG:

[I Bsc](#) [I BBA](#)

[II Bsc](#) [II BBA](#)

[III Bsc](#) [III BBA](#)

PG:

[I MSc](#) [II MSc](#)



When the user is logged in and chooses to give feedback he chooses one class to give feedback. When he chooses to give the feedback, he gets the list option of the semesters to select semester to give the feedback on students.

Similarly for all the classes, it gives the list of semesters to select and to give feedback on the students.

Enter Regdno:

**Get Result**

**Go back to Home ?**

[HOME](#)

The above image shows the asking regdno for the input, when the user enters the regdno and clicks the submit the data of that student is fetched from the database and the details of that student is kept on the form .

Also user has the option to go back to the homepage if the teacher is not interested to give feedback at that time.

This helps the teacher to reduce his work on entering the details of the student, which helps him in saving his prime time.

# Feedback Form :

Campus: MDH

## Feedback from the Department(HOD/Faculty):

Regdno: 204203 Name: Shravan

Class: III UG

Papers	UCSH-501	UCSH-502	UCSH-503	UCSH-504	UCSH-505
Internal CIE	9	9	9	9	9
Marks 70	45	45	45	45	45

## Feedback :

### Faculty :

Faculty:

Warden:

Signature of Parents      Signature of Warden

Signature of Faculty      Signature of Director

sSs

when the teacher enters the Regdno and clicks on submit button, the above feedback form comes on to the page with that particular student details of that particular semester with his Regdno, Name, Class, and marks of that semester including internal CIE and overall marks for 70.

Then the teacher has two textboxes where he can give the feeback to that particular student.

The two textboxes are for the Faculty and Warden. When these two textboxes are filled with feedback the teacher has the option tio print the page and the submits the form to the database.

When the user submits the form to the database, only the data in the two text boxes given will be saved to the database at the student table of that particular student.

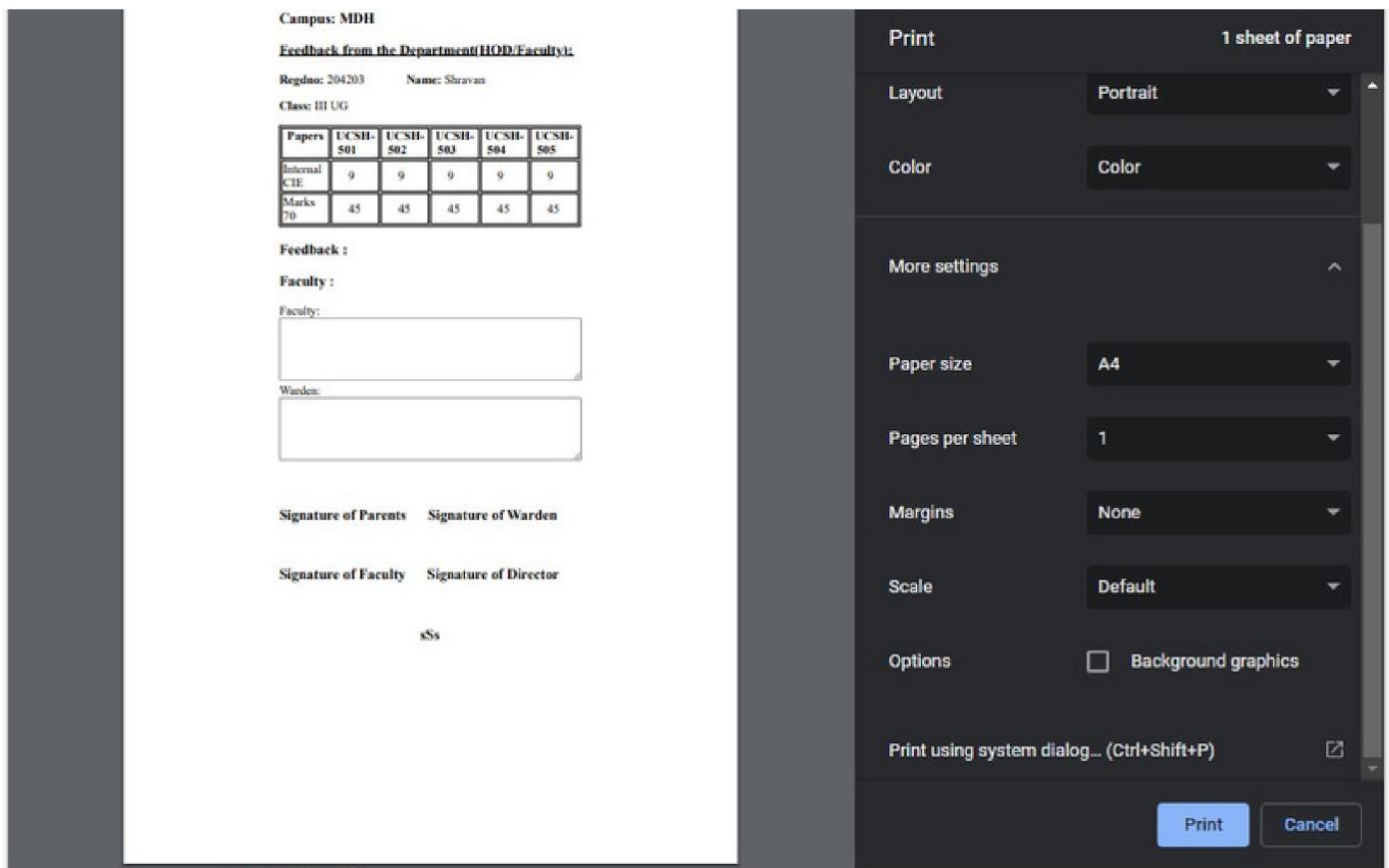
## ***Feedback by Warden :***

When the teacher gives the feedback it gets saved in the database. Then Warden plays his role in giving feedback to the students after the teacher gives.

Now when the warden logins, he will be redirected to the warden's homepage where everything is similar to teacher's portal.

Then when warden selects the class to give feedback to the students the result in the form is the details of the student fetched from the database including the faculty feedback.

By seeing the faculty feedback on the students warden gives his feedback and takes the print. Then warden submits back to the database.



## Print Page :

when the User clicks on print button it also gives the form preview to check the details enters are correct are not in the form. In the end the user has the option to print the page with all the details and issue the form to the student.

The teacher also has the option to save the form as the pdf.

## Uploading Data :

When the teacher logins he also has the option to upload the data of the students to the database. When they upload the data of the students, they get saved in their particular semester table.

The teacher uploaded data helps us to fetch the details of the student from the database and keep it in the form. The teacher uploads the data which consists the fields like Regdno, Name, Class and marks of that semester in which student is studying.

The screenshot shows a portion of a web application interface for a Teacher's Portal. At the top left, it says "Sri Sathya Sai Institu". Below that, a grey bar displays "welcome to Teacher's Portal" and "Logout". In the center, there is a modal dialog titled "Upload Excel File". Inside the dialog, there is a "Select file:" label followed by a "Choose File" button and a message "No file chosen". To the right of the button is an "Upload" button. At the bottom of the dialog, there are two sections: "Please select the classes to give feedback for the students :" and "Please select the classes to upload the data :". Each section contains four dropdown menus labeled "I Bsc", "I BBA", "II Bsc", "II BBA", "III Bsc", "III BBA", "I MSc", and "II MSc". At the very bottom of the page, outside the dialog, is a copyright notice: "© 2020 Copyright: SSSIHL".

In uploading data Section we have the independent classes to upload the data. When the teacher selects one class to upload the data and clicks on it the modal pop-ups and asks the user to select the file to upload. when user selects the file and uploads it gets saved to the database at the student table of that particulat class chosen.

The teacher uploads the data in an excel form.

# Django Admin :

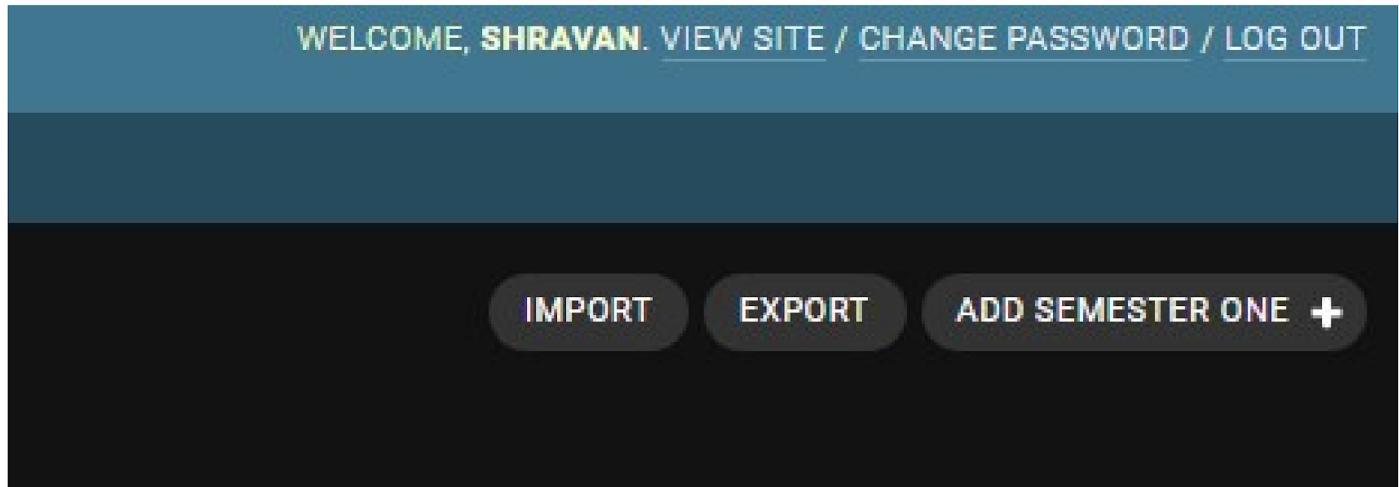
The screenshot shows the Django Admin interface. At the top, there's a header bar with "Django administration" on the left and "WELCOME, SHRAVAN. VIEW SITE / CHANGE PASSWORD / LOG OUT" on the right. Below the header is a sidebar on the left containing a tree view of the database structure. The main area is titled "Recent actions" and lists "My actions" with various entries. The sidebar includes sections for AUTHENTICATION AND AUTHORIZATION, MAIN, and STUDENTS.

Model	Action	Details
Groups	+ Add	
Users	+ Add	
<hr/>		
MAIN		
psc fours	+ Add	Change
psc ones	+ Add	Change
psc threes	+ Add	Change
psc twos	+ Add	Change
semester fours	+ Add	Change
semester ones	+ Add	Change
semester sixs	+ Add	Change
semester threes	+ Add	Change
semester twos	+ Add	Change
Students	+ Add	Change
Studentss	+ Add	Change

Django admin is like the admin for the whole project. There is a superuser created who is called as admin. He checks all the information in the databases are stored properly or not, and also has the many operations to perform inside the database.

Django-admin is Django's command-line utility for administrative tasks. This document outlines all it can do. In addition, `manage.py` is automatically created in each Django project.

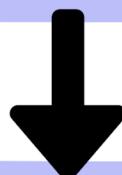
The django admin also has the feature to import or export different file formats within the admin application.



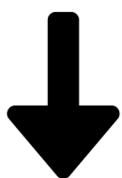
The Django admin application can use your models to automatically build a site area that you can use to create, view, update, and delete records. This can save you a lot of time during development, making it very easy to test your models and get a feel for whether you have the right data. The admin application can also be useful for managing data in production, depending on the type of website. The Django project recommends it only for internal data management (i.e. just for use by admins, or people internal to your organization), as the model-centric approach is not necessarily the best possible interface for all users, and exposes a lot of unnecessary detail about the models.

## Approach :

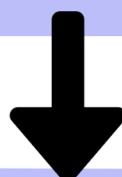
**PROJECT : FEEDBACK GENERATION**



**OUTCOME : FEEDBACK FORM**



**TASKS: UPLOAD DATA/ GIVE FEEDBACK**



**KEY ROLE : TEACHERS ( TO GIVE FEEDBACK)**



**TIMELINE**

## **Feed back generation:**

feed back form is all about feedback form that allows students to submit their feedback about a semester, including their registration number, name, class, and marks for different subjects. The form also includes two text areas where students can provide feedback for faculty and warden.

The form is built using HTML and Django, which allows for dynamic content to be generated based on the data provided by the user. The form uses a "POST" method to submit the data, and includes a "CSRF" token to protect against cross-site request forgery attacks.

Overall, the feedback form is designed to collect feedback from students and allow them to provide their opinions about the semester and the teaching staff, which can be used to improve the quality of education and services provided by the institution.

fetching the data provided by the course of teaching faculty based in the internal assignment conducted in course span on individual ,by taking all this data from all papers conducted in to excel form based on the students roll number.

## **Out come:**

out come of the feed back form is based on the students roll number from institute it contains all internal marks and and feed back from

faculty and warden it also contain the option for parents also to give feed back form and signature and this help us to

## **Upload Data :**

To upload data from an Excel sheet into a Django form, you can follow these steps:

Save the Excel sheet in a CSV (Comma Separated Values) format.

In Django, create a Model that represents the data you want to upload from the CSV file.

For example, if you want to upload student marks, create a Student model with fields for name, roll number, and marks.

Create a Django Form that uses the Model created in step 2. This Form will be used to create new Student objects from the data in the CSV file.

In your Django view, write a function that will read the CSV file and create new Student objects using the Form created in step 3. In your Django template, create a file input field and a submit button. When the user selects a CSV file and clicks the submit button, the view created in step 4 will be called to process the CSV data and create new Student objects.

There are various libraries and packages available in Python to help with CSV processing and uploading data into Django forms, such as pandas and Django's built-in CSV module.

## Teachers Role:

Create a Django Model for student marks with fields for student name, roll number, and marks in different subjects.

Create a Django Form for the teacher to upload a CSV file containing student marks. This form will be used to create new objects in the database using the Model created in step 1.

1. Assign appropriate permissions to the teacher role, such as permission to add, change, and view student marks, but not delete or edit other data.
2. Create a Django View that handles the file upload and creates new objects in the database using the Form created in step 2.
3. When the teacher uploads the CSV file using the form, the View created in step 4 will be called to process the data and create new objects in the database.
4. To verify the data, the teacher can use Django's Admin interface or a custom Django View to view and edit the data. The teacher can also create custom filters and search options to find specific data quickly.
5. If the teacher finds any errors in the data, they can use the permissions assigned to their role to edit the data or delete incorrect records.

## **Conclusion :**

The final conclusion of our project is to help the teachers to give the feedback to the students without having much of the work they are doing. This project helps them to reduce the work load on them without actually entering the data of all the students.

By utilizing feedback forms, teachers can gain valuable insights into the learning process of their students, identify areas for improvement

Moreover, feedback forms empower students to voice their opinions and concerns about their education, which in turn can lead to a more engaging and collaborative learning environment.

This project also helps them to reduce the manual work which takes their lot of energy in entering the details of the students.

This project also helped us to learn many more topics which helps us in knowing more technical terms and using them when needed.

However, this does not mean that our web application cannot grow. we personally look at this web application as Automation of Feed back form version 1.

Therefore, universities should prioritize the use of feedback forms as part of a comprehensive feedback system that fosters continuous improvement and growth for all stakeholders involved.

## Bibliography :

Django Documentation ---

<https://docs.djangoproject.com/en/4.2/>

<https://www.techwithtim.net/tutorials/django/>

User Login :

<https://docs.djangoproject.com/en/4.2/topics/auth/>

Forms :

<https://docs.djangoproject.com/en/4.2/topics/forms/>

Bootstrap :

<https://www.w3schools.com/bootstrap5/index.php>

