### **A PROJECT REPORT**

**STUDENT MANAGEMENT SYSTEM**

***Submitted by***

### **AYUSH SHRIVASTAVA [I]-(2115000266)**

### **DARSHAN GUPTA [A]-(2115000314)**

**VARDHAN GUPTA [C]-(2115001094)**

***in partial fulfillment for the award of the degree of***

### **BACHELOR OF ENGINEERING**

**IN**

#### Department of Computer Science & Engineering



**GLA University, Mathura**

##### 

NOVEMBER 2023

#### **BONAFIDE CERTIFICATE**

Certified that this project report **“STUDENT MANAGEMENT SYSTEM”** is the bonafide work of “**AYUSH SHRIVASTAVA, DARSHAN GUPTA, VARDHAN GUPTA”** who carried out the project work under my supervision

|  |  |
| --- | --- |
|  |  |
| **SIGNATURE**  Mr. Rohit Agarwal  **HEAD OF THE DEPARTMENT**  Computer Engineering & Applications | **SIGNATURE**  Mr. Suman Kumar Das  **SUPERVISOR**  Technical Trainer  Computer Engineering & Applications |

Submitted for the project viva-voce examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# **List of Figures**

# Figure 3.1 Steps to Create Web Site………………..............................................

Figure 3.2 Login form Without CSS………………………………………………….

Figure 3.3 Login form With CSS………………………………………………………

Figure 3.2.1 Programming Language popularity……………………………………

Figure 3.2.2 Scripting…………………………………………………………………..

Figure 4.1.1 Django – Python………………………………………………………….

Figure 5.1 DFD…………………………………………………………………………..

Figure 5.1.1 DFD-1……………………………………………………………………….

Figure 5.2.1 DFD-2……………………………………………………………………….

Figure 6.1.1 Login Page…………………………………………………………………

Figure 6.1.2 Front Page………………………………………………………………….

Figure 6.1.3 Dashboard………………………………………………………………….

Figure 6.1.4 HOD Portal………………………………………………………………….

Figure 6.1.5 Staff Portal…………………………………………………………………..

Figure 6.1.6 Student’s Portal……………………………………………………………..

Figure 6.1.7 Staff Leave……………………………………………………………………

Figure 6.1.8 Result Section……………………………………………………………….

Figure 6.1.9 Feedback Page………………………………………………………………

CHAPTER - I

#### **INTRODUCTION**

In the ever-evolving landscape of education, the efficient management of student-related information is paramount to the success of educational institutions. The advent of technology has ushered in a new era of possibilities, providing opportunities to streamline administrative processes, enhance communication, and optimize the overall learning experience. This project aims to introduce and implement a comprehensive Student Management System tailored to meet the specific needs and challenges faced by our educational institution.

This project holds the promise of not only modernizing our educational administration but also fostering a collaborative and engaging learning environment. By leveraging technology strategically, we aim to propel our institution into a future where administrative tasks are streamlined, educators are empowered, and students are equipped with the tools they need for academic success. Through the implementation of the Student Management System, we embark on a journey towards a more connected, efficient, and innovative educational ecosystem.

## **1.1 OBJECTIVE**

The primary theoretical objectives of implementing a Student Management System (SMS) project within an educational institution encompass a holistic improvement in administrative efficiency, communication dynamics, and the overall learning experience.

These objectives provide a foundation for the practical implementation of the project, aligning educational technology with established theories to create a system that harmonizes with pedagogical principles and organizational needs

CHAPTER-2

**THE STEPS TO CREATE A WEB SITE**

Creating a web site requires multiple steps which includes the following:

* Creating a UI(User Interface)
* Scripting(Both at server and client end)
* Creating a backend or the database

**2.1 UI DEVELOPMENT**

Technologies that are mostly used to develop a User Interface are:

* HTML
* CSS
* Bootstrap

2.1.1 HTML

HTML, which stands for Hypertext Markup Language, is the standard markup language used to create and structure content on the World Wide Web. It provides a set of elements or tags that define the structure of a web page, indicating how different pieces of content are organized and displayed within a browser.

HTML is the foundational language for creating web content. It works in conjunction with CSS (Cascading Style Sheets) for styling and layout, and JavaScript for dynamic behavior, collectively forming the backbone of modern web development.

General Syntax of HTML

<! DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>My First Heading</h1>  
<p>My first paragraph</p>  
  
</body>  
</html>

(The text between <html> and</html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text <title> and></title> defines the browser page title)

The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to “quirks mode” for rendering.

2.1.2 CSS

CSS, which stands for Cascading Style Sheets, is a style sheet language used for describing the presentation and visual styling of HTML and XML documents. It enables web developers to control the layout, appearance, and formatting of elements on a web page. CSS allows for the separation of content from presentation, promoting cleaner and more maintainable code.

CSS is a crucial technology in web development, enabling developers to create visually appealing and responsive user interfaces. It works in tandem with HTML and JavaScript to build dynamic and engaging web applications. The cascading nature of CSS allows styles to be applied hierarchically, providing flexibility and control over the appearance of web content.

Cascading Style Sheet (CSS) is used to set the style in web pages that contain HTML elements. It sets the background color, font-size, font-family, color, etc. properties of elements on a web page.

There are three types of CSS which are given below:

* Inline CSS - Inline CSS contains the CSS property in the body section attached to the element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute. Example: This example shows the application of inline-css.

General Syntax of Inline CSS

<!DOCTYPE html>

<html>

<head>

<title>Inline CSS</title>

</head>

<body>

<p style="color:#009900; font-size:50px;

font-style:italic; text-align:center;">

GeeksForGeeks

</p>

</body>

</html>

* Internal or Embedded CSS - This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e. the CSS is embedded within the <style> tag inside the head section of the HTML file.

General Syntax of Internal CSS

<!DOCTYPE html>

<html>

<head>

<title>Internal CSS</title>

<style>

.main {

text-align: center;

}

.GFG {

color: #009900;

font-size: 50px;

font-weight: bold;

}

.geeks {

font-style: bold;

font-size: 20px;

}

</style>

</head>

<body>

<div class="main">

<div class="GFG">GeeksForGeeks</div>

<div class="geeks">

A computer science portal for geeks

</div>

</div>

</body>

</html>

* External CSS - External CSS contains separate CSS files that contain only style properties with the help of tag attributes (For example class, id, heading, etc). CSS property is written in a separate file with a .css extension and should be linked to the HTML document using a link tag. It means that, for each element, style can be set only once and will be applied across web pages.

General Syntax of External CSS

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="geeks.css" />

</head>

<body>

<div class="main">

<div class="GFG">GeeksForGeeks</div>

<div id="geeks">

A computer science portal for geeks

</div>

</div>

</body>

</html>

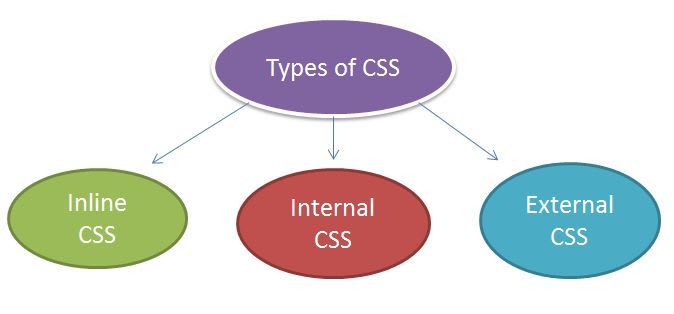


Fig 2.1

2.1.3 BOOTSTRAP

Bootstrap is an open-source front-end framework developed by Twitter. It provides a collection of pre-designed and pre-built components, styles, and JavaScript plugins that facilitate the development of responsive and visually appealing web pages and web applications. Bootstrap is based on HTML, CSS, and JavaScript, and it aims to simplify the process of creating consistent, responsive, and mobile-friendly user interfaces.

**2.2 SCRIPTING**

There are two scripting methodologies:

1. Server side scripting: This scripting is done at the server end.

2. Client side scripting: This is done at the client end or the browser.

2.2.1 SERVER SIDE SCRIPTING

Server-side scripting is a programming technique where scripts are executed on a web server to generate dynamic web pages. Unlike client-side scripting, which runs in the user's browser, server-side scripts are processed on the server before the content is sent to the client's browser. This allows for the generation of dynamic content, interaction with databases, and the execution of server-side logic.

Here's a simple example of a server-side scripting using Django. Django is a Python web framework that encourages rapid development and clean, pragmatic design.

There are several languages that can be used for server side programming :

* PHP
* ASP.NET(C# OR Visual Basic)
* C++
* Java and JSP
* Python
* Ruby on Rails and so on.

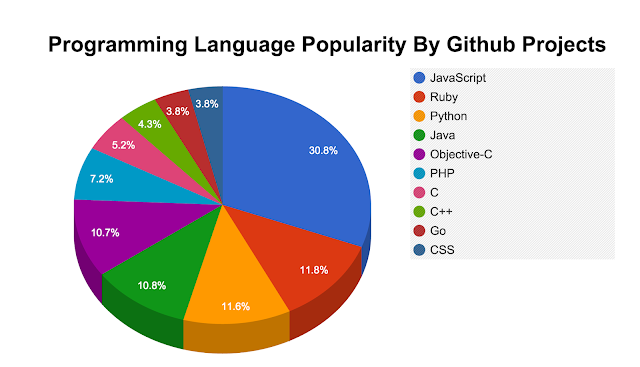
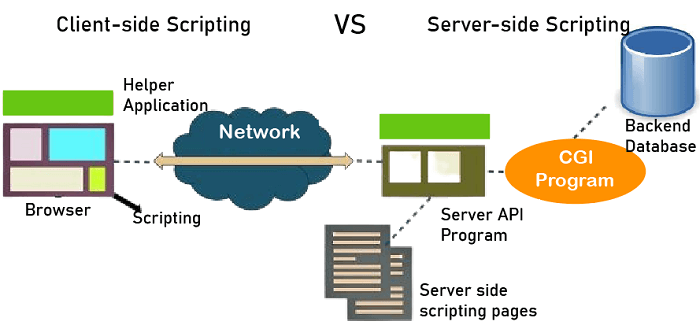


Fig 2.2

2.2.2 CLIENT SIDE SCRIPTING

Client-side scripting refers to scripts or code that run on the user's web browser rather than on the web server. This type of scripting is commonly used to enhance the user interface and interactivity of a website. The most common languages for client-side scripting are JavaScript, HTML, and CSS.

It's important to note that while client-side scripting enhances the user experience, security measures must be in place on the server-side to validate and process data, as client-side code can be manipulated by users.



CHAPTER-3

**SCRIPTING LANGUAGES**

**3.1 DJANGO**

Django is a Python framework that makes it easier to create web sites using Python.

Django takes care of the difficult stuff so that you can concentrate on building your web applications.

Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

****

FIG 3.1

In Django, the database is a crucial component for storing and retrieving data. Django's Object-Relational Mapping (ORM) system allows developers to interact with the database using Python code rather than raw SQL queries.

**3.2 INSTALLING DJANGO**

Django is installed using pip, with this command:

****

Fig 3.2

Which will give a result that looks like this:

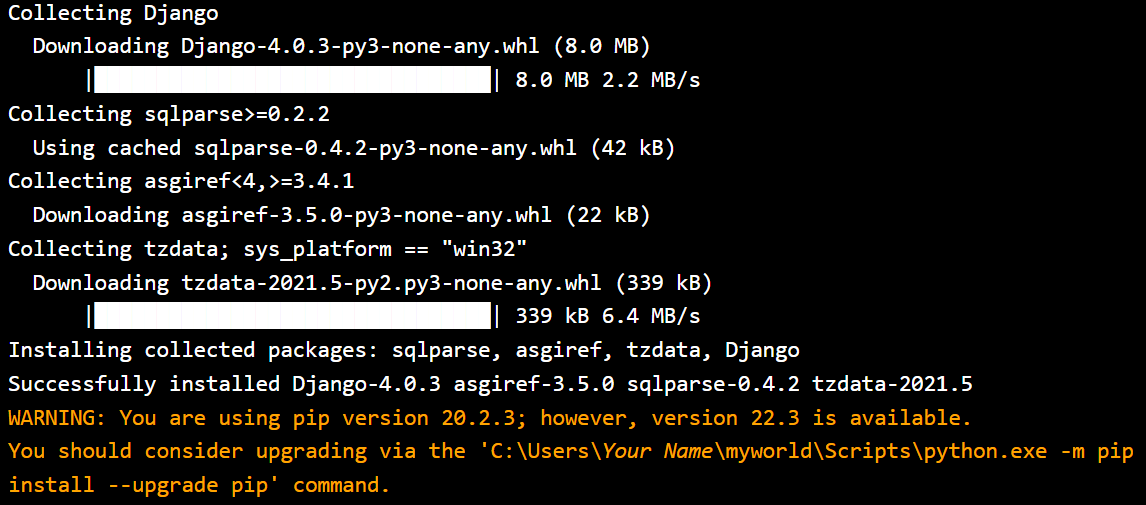


Fig 3.3

Now , check the python version:



Fig 3.4

Check Django version:



Fig 3.5

Django project in a virtual environment on your computer.

**3.3 JAVASCRIPT**

JavaScript (JS) is the most popular lightweight, interpreted compiled programming language. It can be used for both Client-side as well as Server-side developments. Java Script also known as a scripting language for web pages. This JavaScript Tutorial is designed to help both beginners and experienced professionals master the fundamentals of Java Script and unleash their creativity to build powerful web applications .From basic syntax and data types to advanced topics such as object-oriented programming and DOM manipulation.

JavaScript can be added to HTML file in two ways:

* Internal JS: We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
* External JS: We can write Java Script code in another files having an extension.js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

**3.4 JQuery**

JQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT license. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

JQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the JQuery library allows the creation of powerful dynamic web pages and Web applications.

The set of JQuery core features-DOM element selections, traversal and manipulation-enabled by its selector engine (named "Sizzle" from v1.3), created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API.

Microsoft and Nokia bundle jQuery on their platforms. Microsoft includes it with Visual Studio for use within Microsoft's ASP.NET AJAX and ASP.NET MVC frameworks while Nokia has integrated it into the Web Run-Time widget development platform.

**3.5 FEATURES**

Django is a high-level web framework for building web applications quickly and with a clean, pragmatic design. It follows the Model-View-Template (MVT) architectural pattern, which is a slight variation of the traditional Model-View-Controller (MVC) pattern. Here are some key features of Django:

**1.Object-Relational Mapping (ORM):**

* Django provides a powerful and high-level Object-Relational Mapping system that allows developers to interact with databases using Python classes. Models define the database schema and facilitate database operations.

**2.Admin Interface:**

* Django includes an automatic admin interface that allows developers to manage application data using a web-based interface. It's customizable and generated based on the models defined in the application.

**3. URL Routing:**

* Django uses a URL routing system to map URLs to views. The urls.py file defines the URL patterns, and views handle the logic associated with those patterns.

**4. Views and Templates:**

* Views in Django handle the processing of user requests and return appropriate responses. Templates are used to generate HTML dynamically and separate the presentation logic from the business logic.

**5. Middleware:**

* Middleware components in Django allow developers to process requests and responses globally. This can include authentication, security, and other custom processing.

Django's features are designed to promote rapid development, clean and maintainable code, and adherence to best practices in web development. It's a popular choice for building a wide range of web applications, from small projects to large-scale enterprise applications.

CHAPTER-4

**SOFTWARE REQUIREMENT SPECIFICATION**

**4.1 HARDWARE REQUIREMENTS**

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware , the size and requirements are also important.

|  |  |
| --- | --- |
| Processor | Intel CORE i5 |
| RAM | 4.0GB |
| Hard Disk Drive | 250GB |

**4.2 SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| No. | **Description** |
| 1 | Windows 7,8,10 |
| 2 | HTML , CSS , JavaScript , Bootstrap |
| 3 | Python – Django |
| 4 | Compiler : python manage.py runserver |

CHAPTER-5

**DATA FLOW DIAGRAM**

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

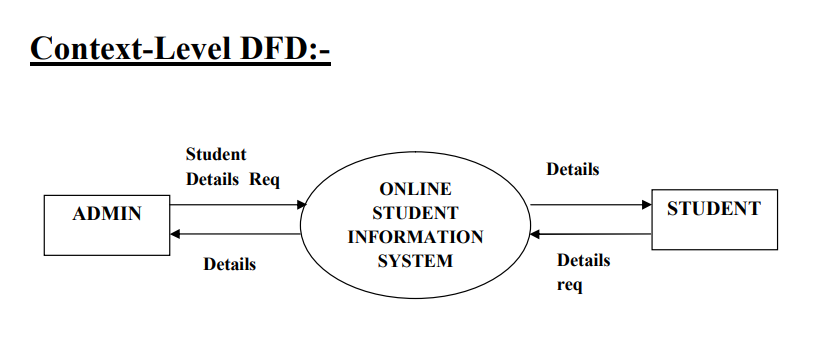
1. Rectangles representing external entities, which are sources or destinations of data.

II. Ellipses representing processes, which take data as input, validate and process it and output it.

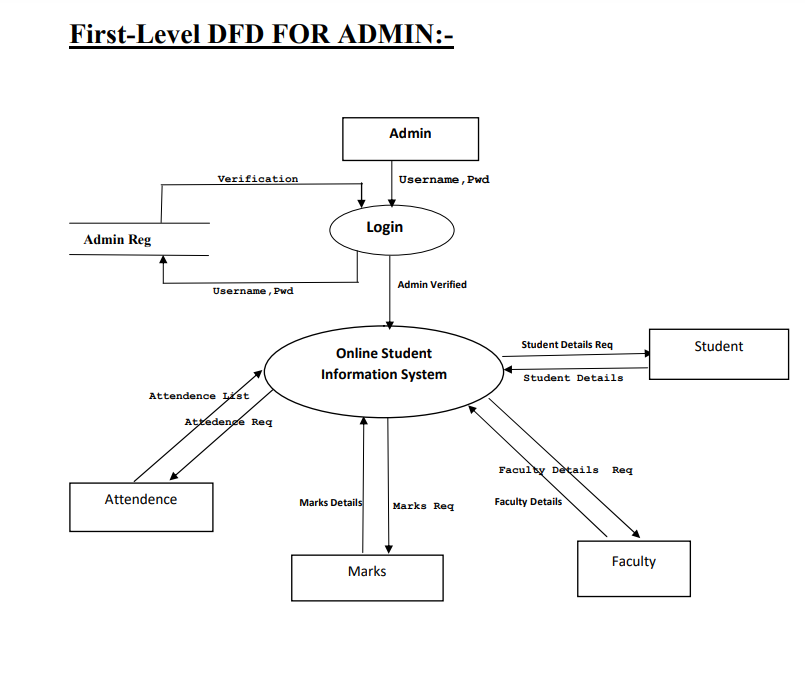
III. Arrows representing the data flows, which can either, be electronic data or physical items.

IV. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

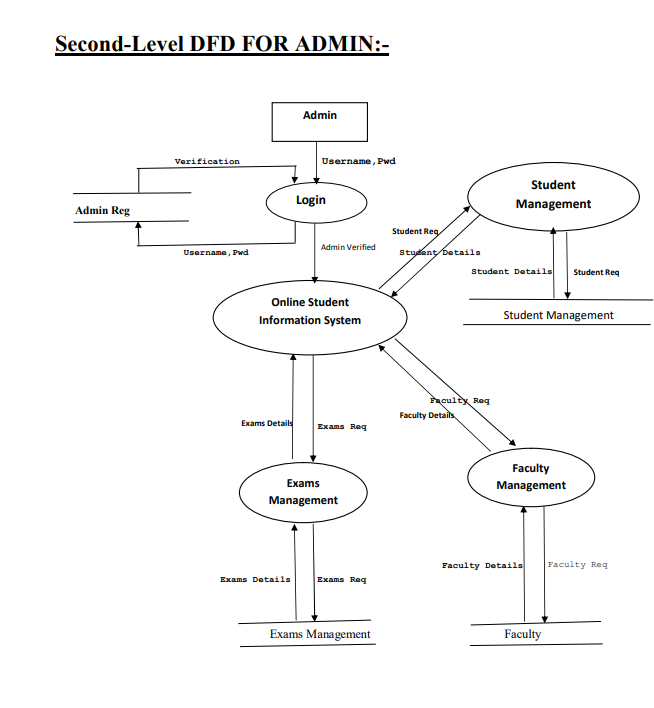
Figures below are the Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input, output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

****

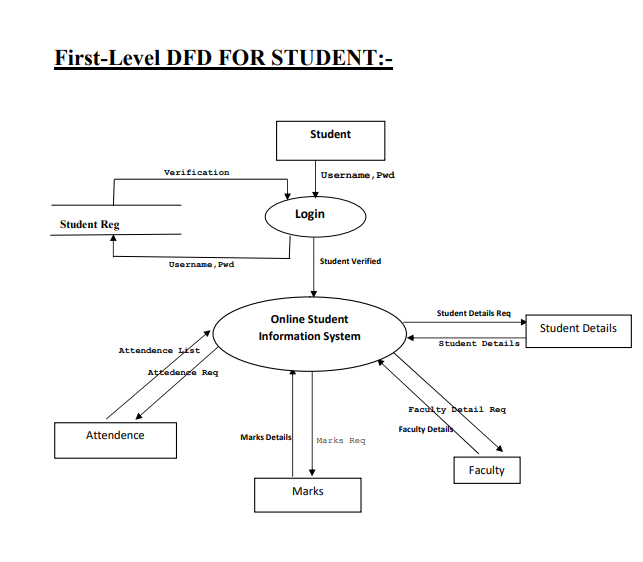
**Fig 6.1**

****

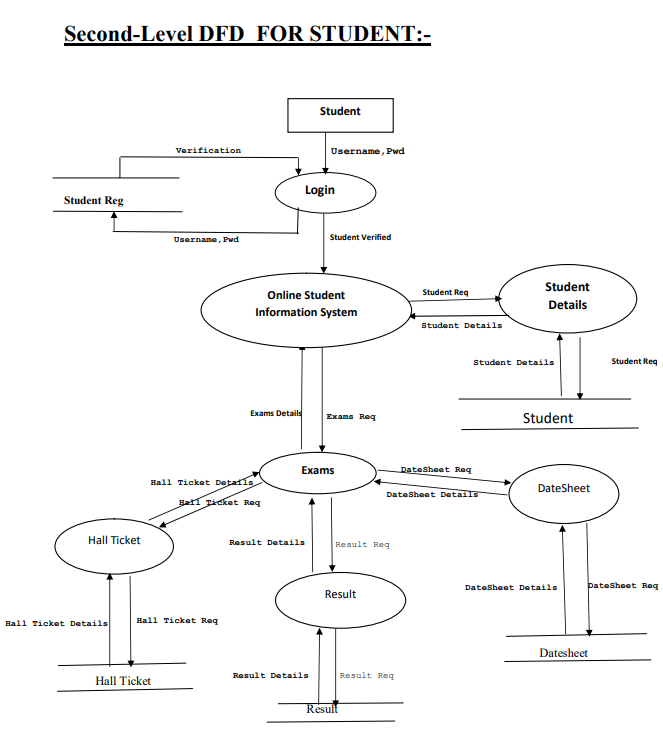
**Fig 6.2**

****

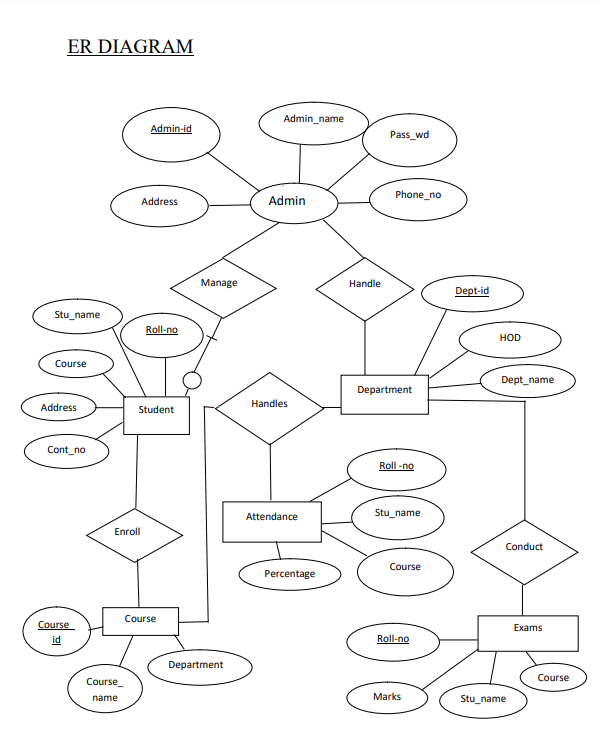
**Fig 6.3**

****

**Fig 6.4**

****

**Fig 6.5**

****

**Fig 6.6**

CHAPTER-6

**PROJECT**

**7.1 PROJECT (Advanced Technologies):**

Name: Fly High

**7.TECHNOLOGIES USED:**

* HTML
* CSS
* Bootstrap
* JavaScript
* JQuery
* Django / Python

Server: python manage.py runserver

Database: DbsqLite(Inbuilt database)

Operating System : Windows 7/8/10

**7.3 TECHNICAL DETAILS**

Front end is designed using HTML, CSS and Bootstrap. used to perform behind the screen requests and JavaScript used to perform client side scripting

➤ Backend is based on Django +Dbsqlite

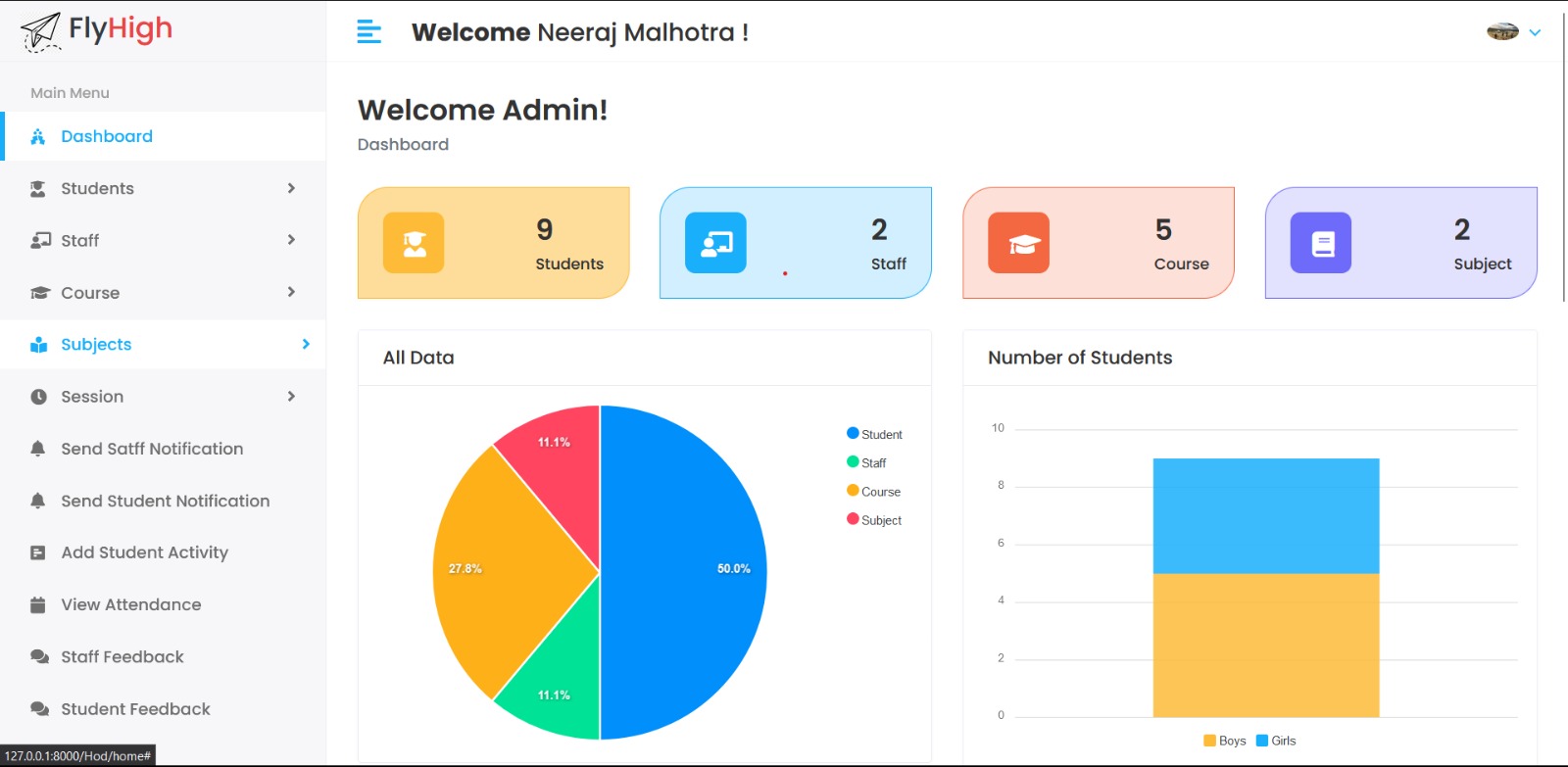
➤ Backend online host includes a centralized database resident on the server, the script which is built in Django used to Dbsqlite query the database on user's request for transaction of data

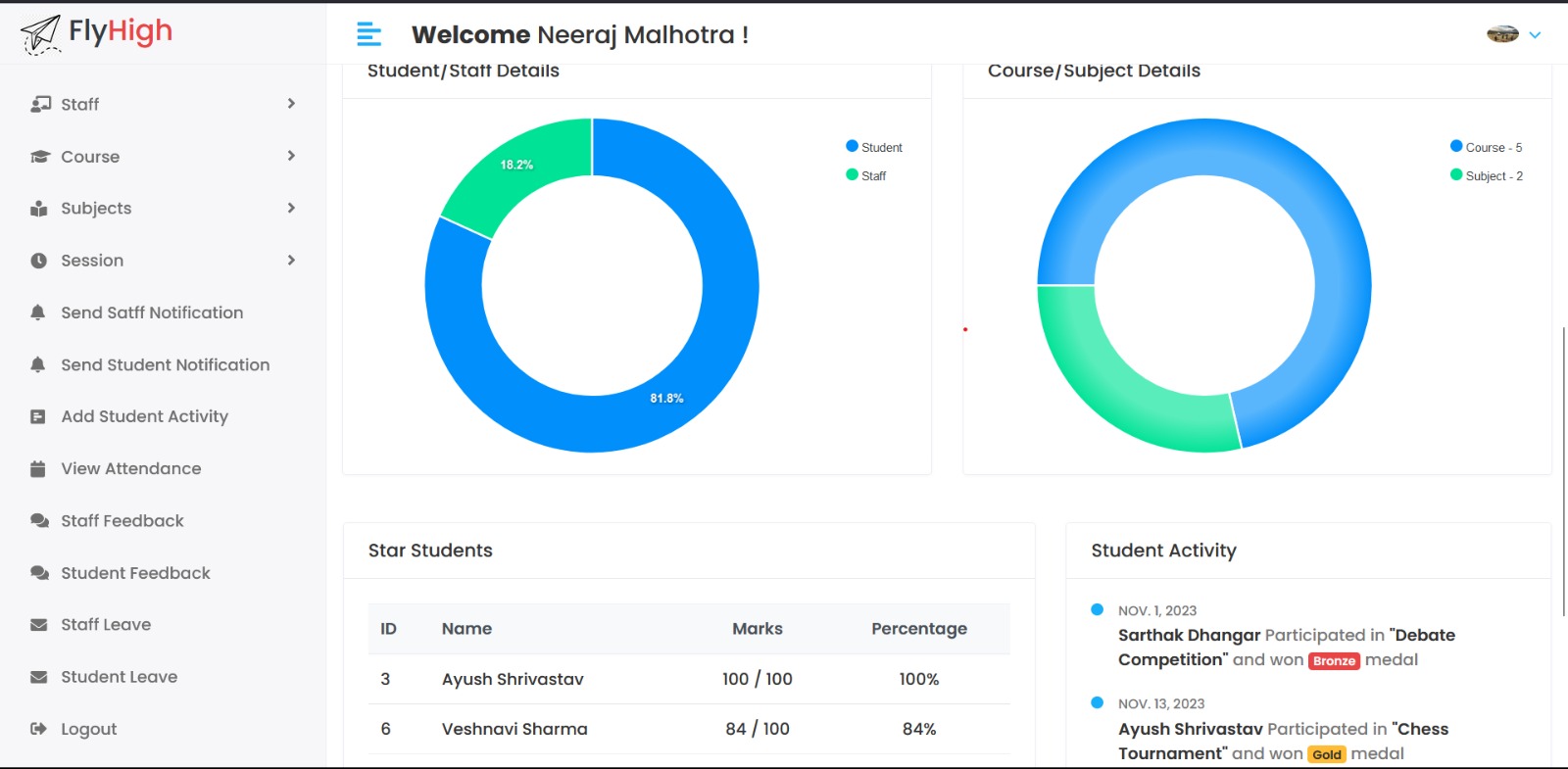
➤ The forms are made using the HTML, Bootstrap for designing and Django for back-end

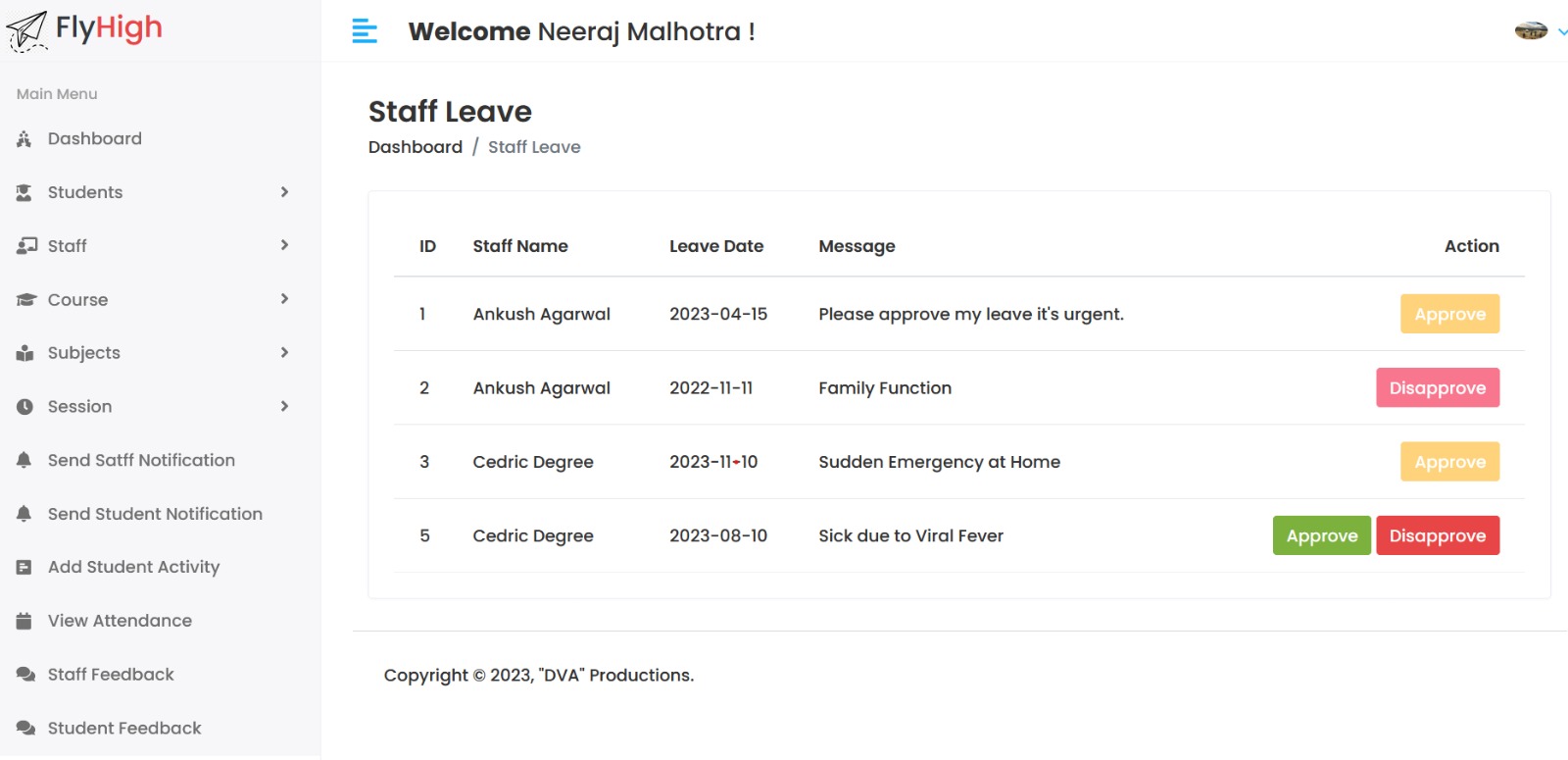
➤ JavaScript and JQuery used for client side scripting and Django for the server side development

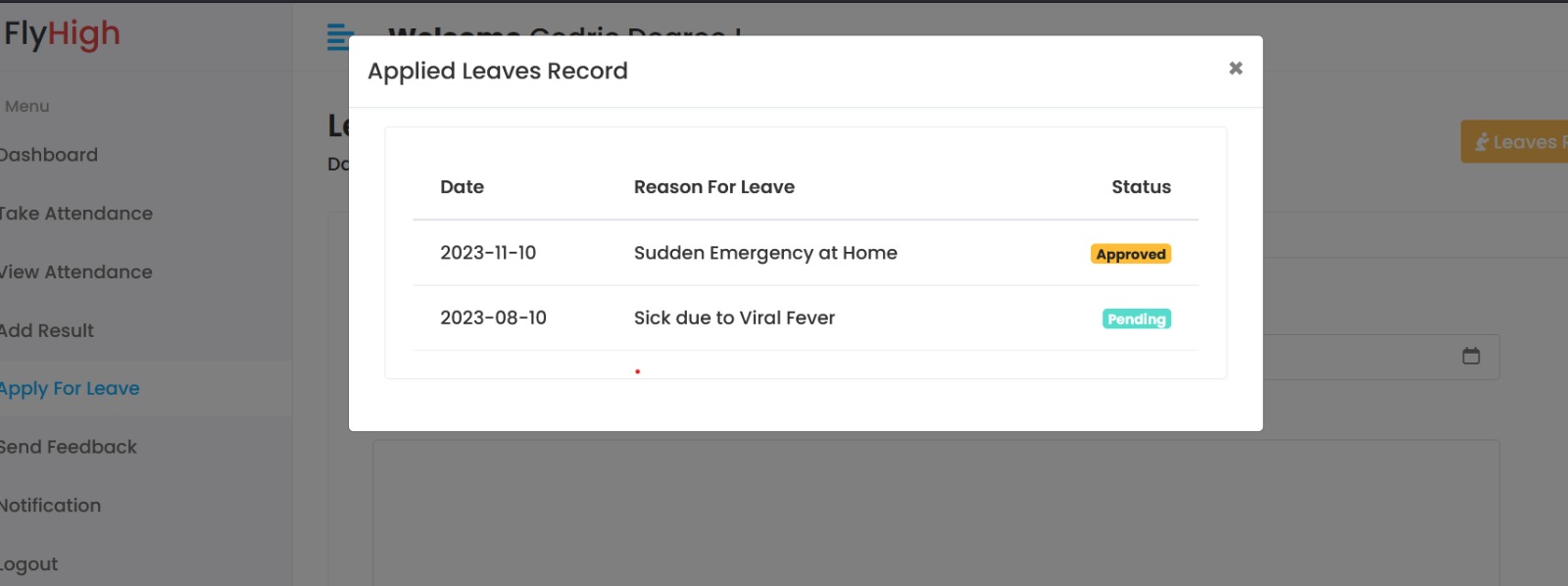
**CHAPTER-8**

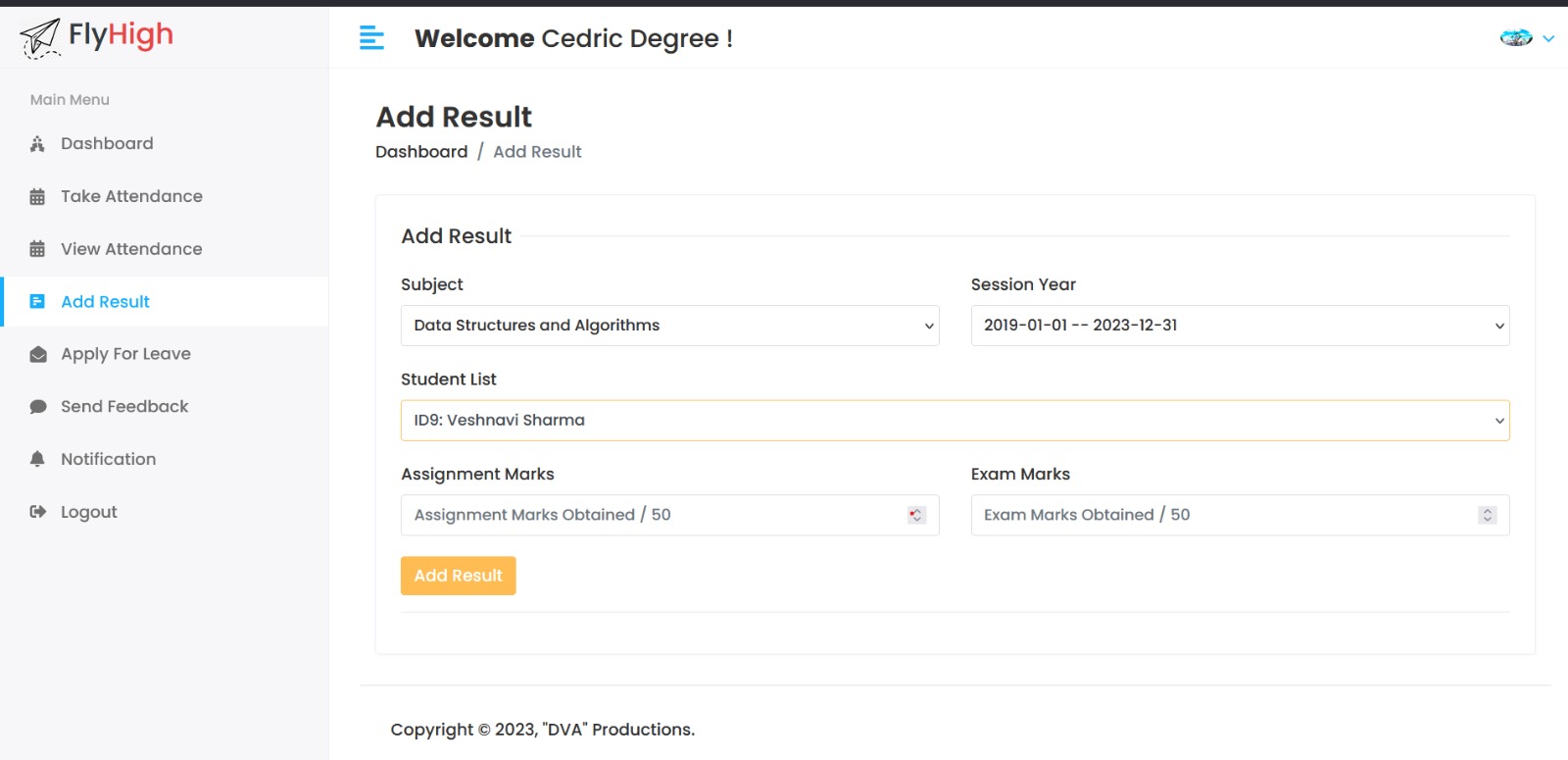
**FRONTEND SCREENSHOTS**

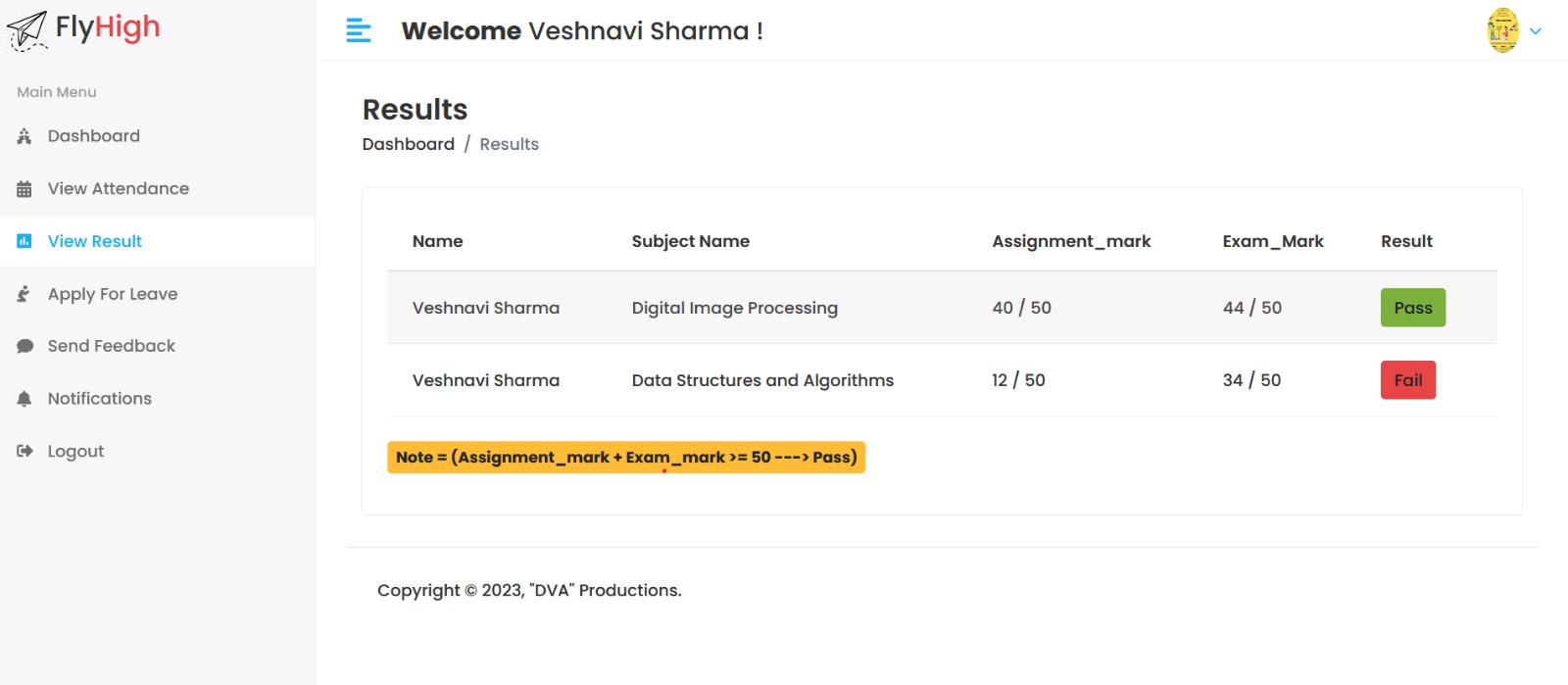
****

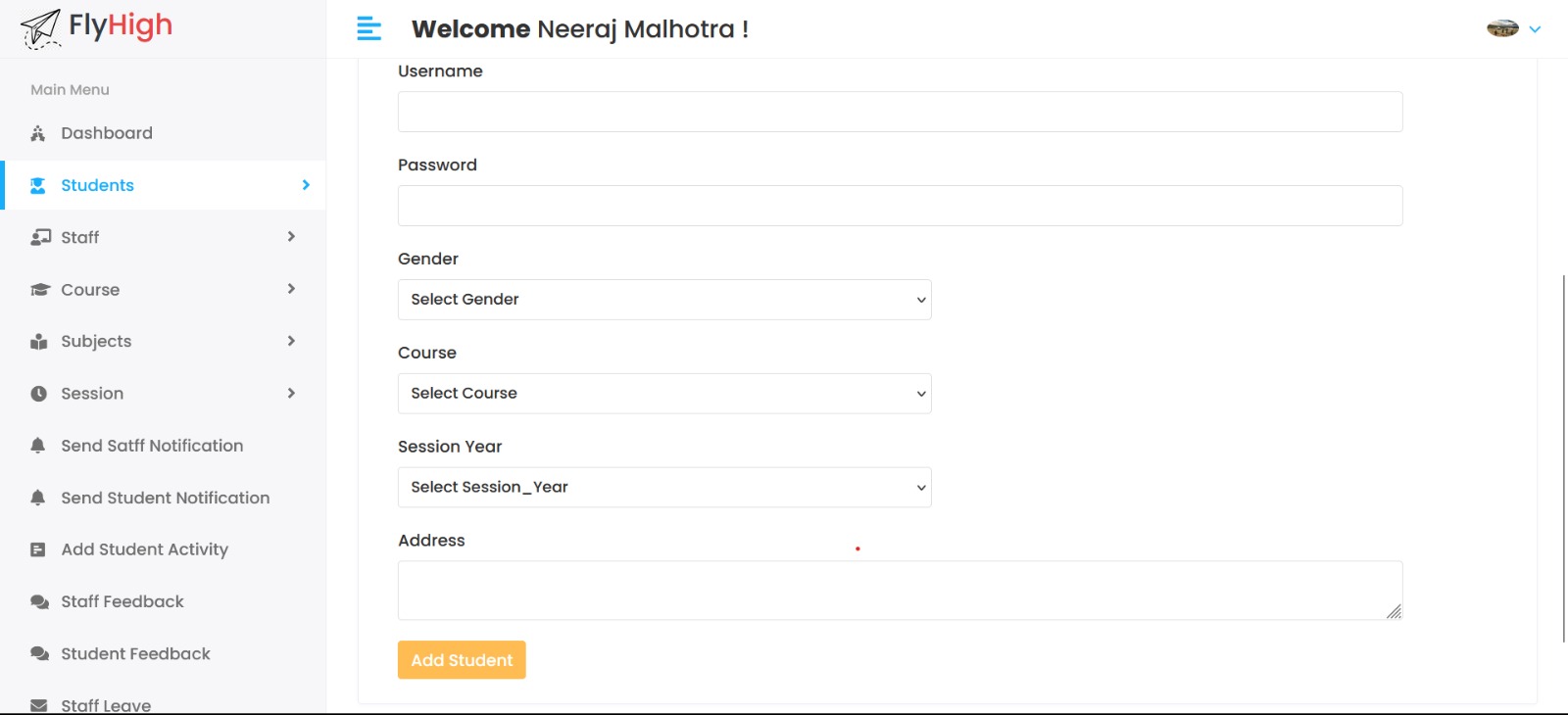
****

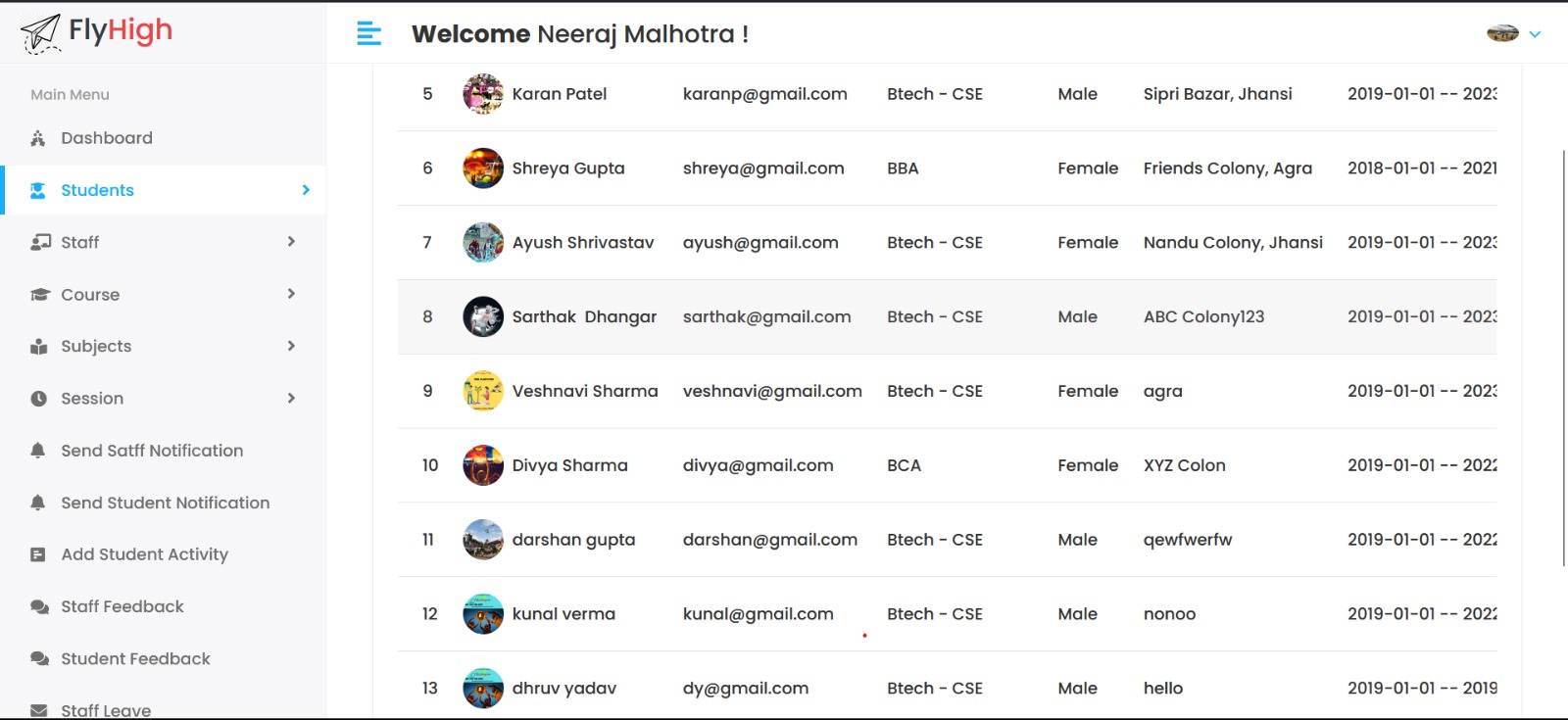
****

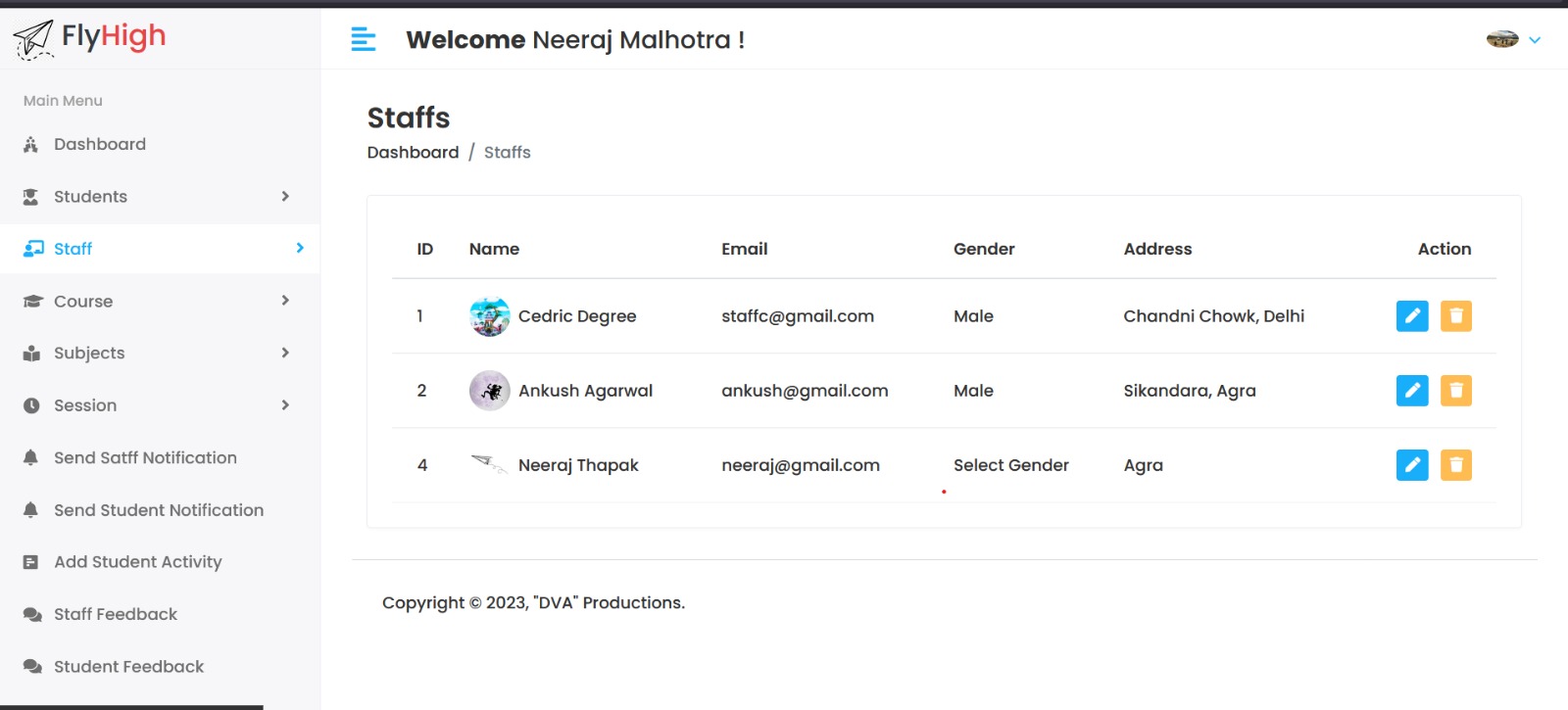
****

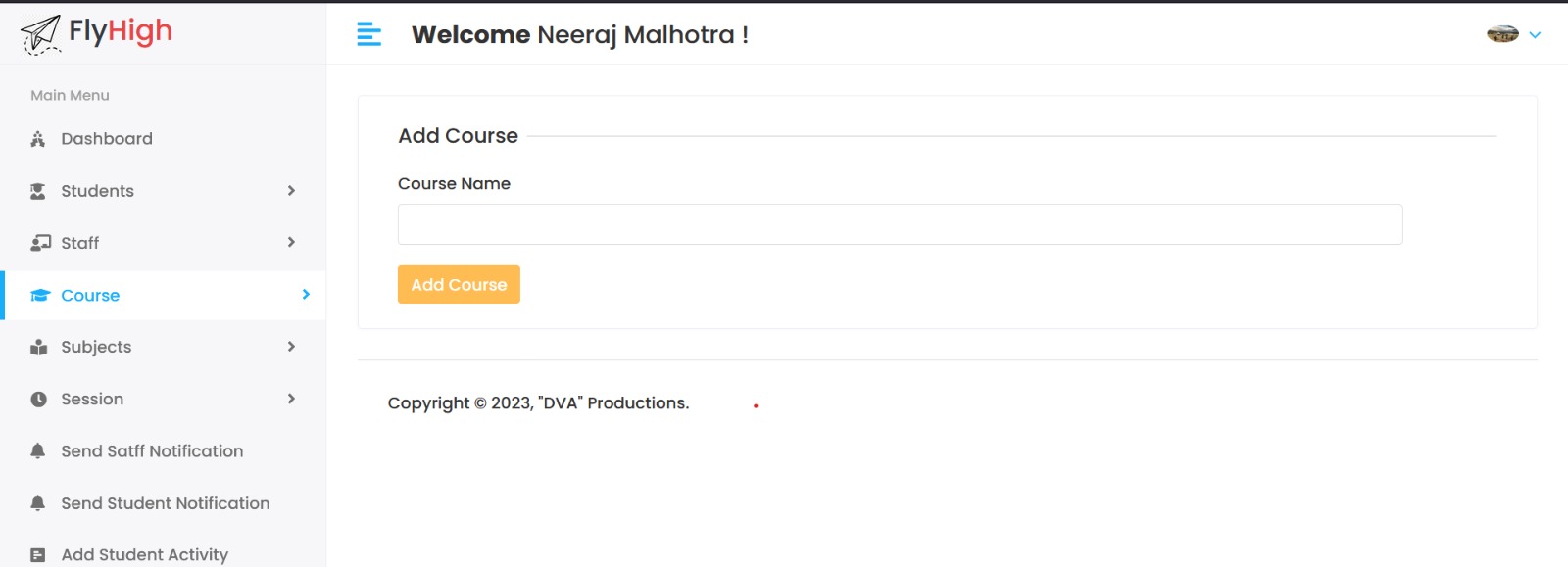
****

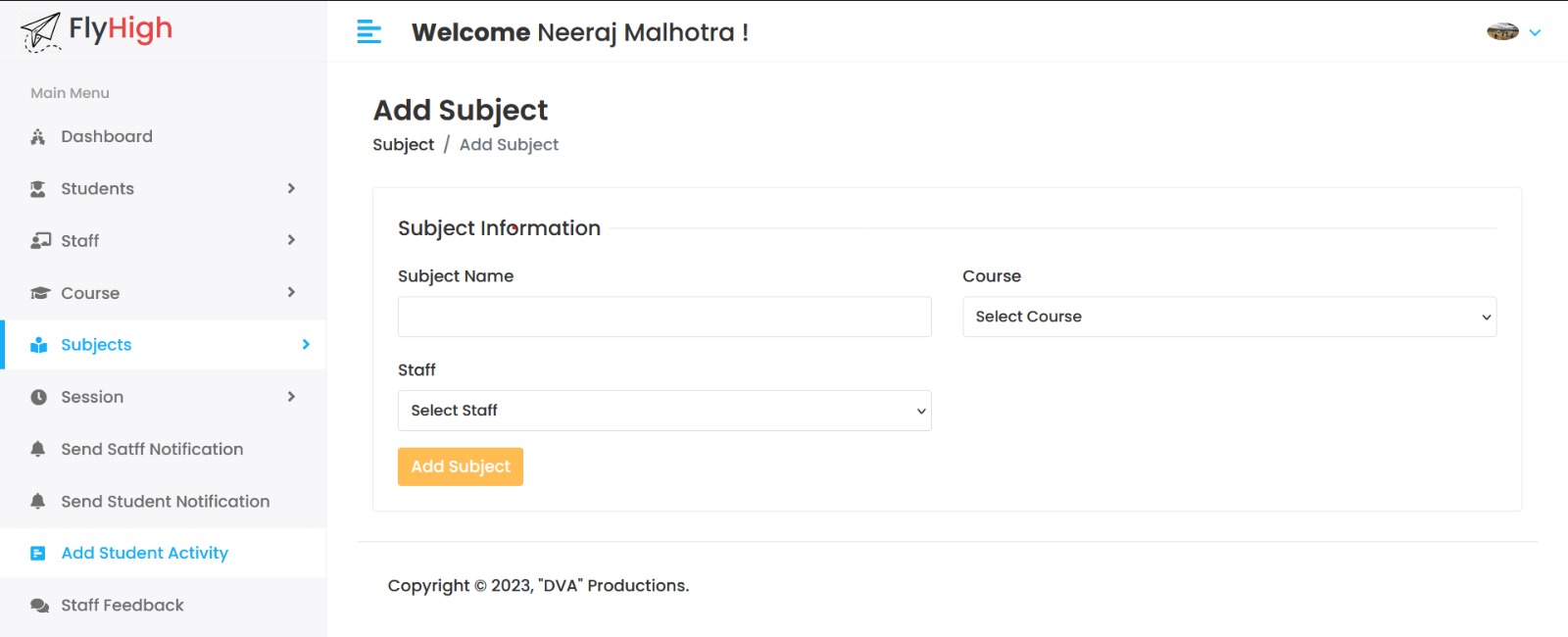
****

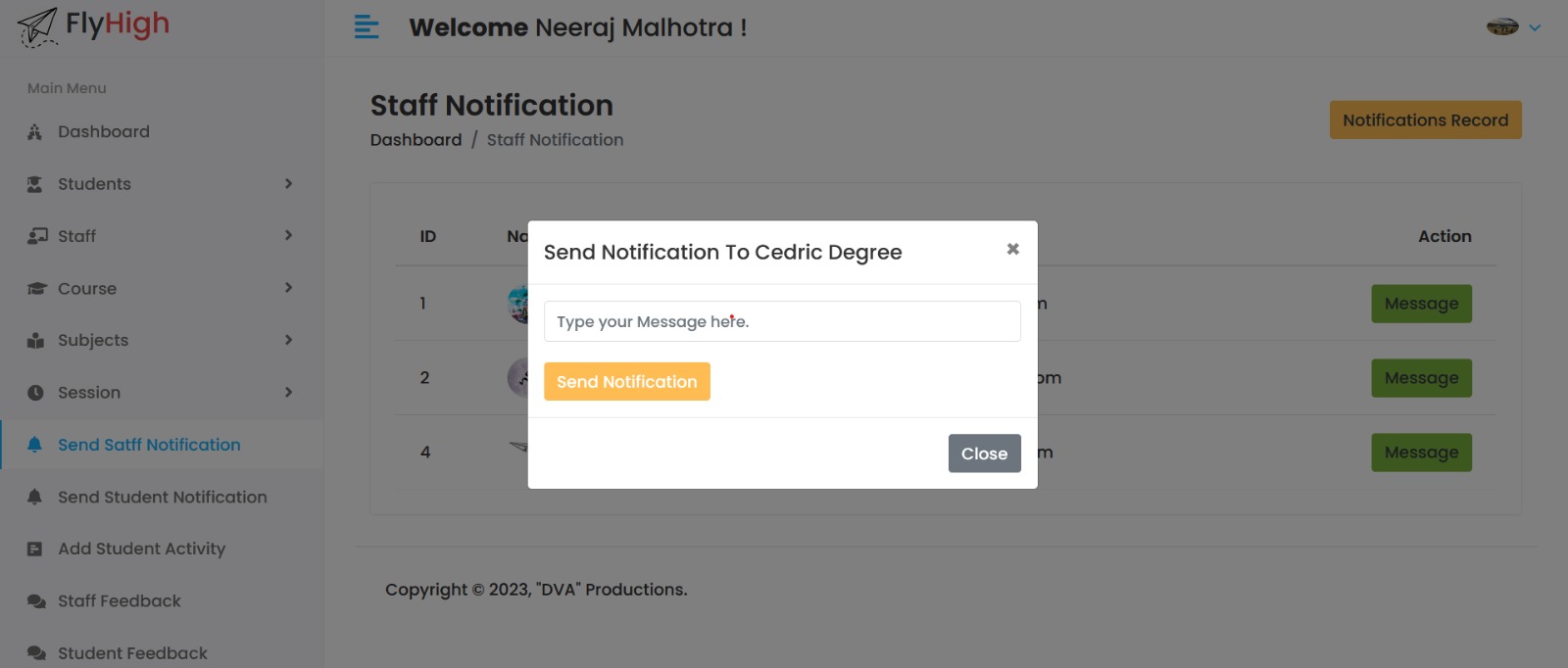
****

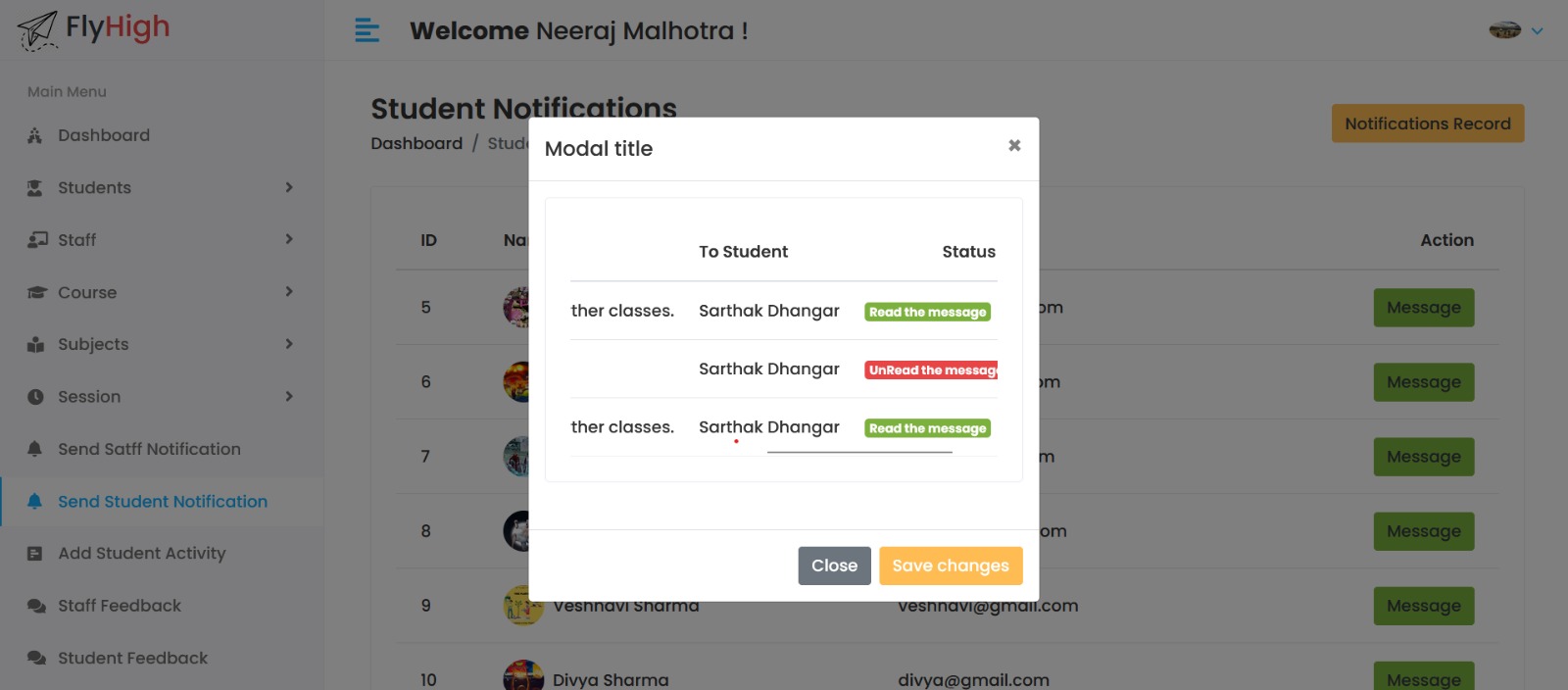
****

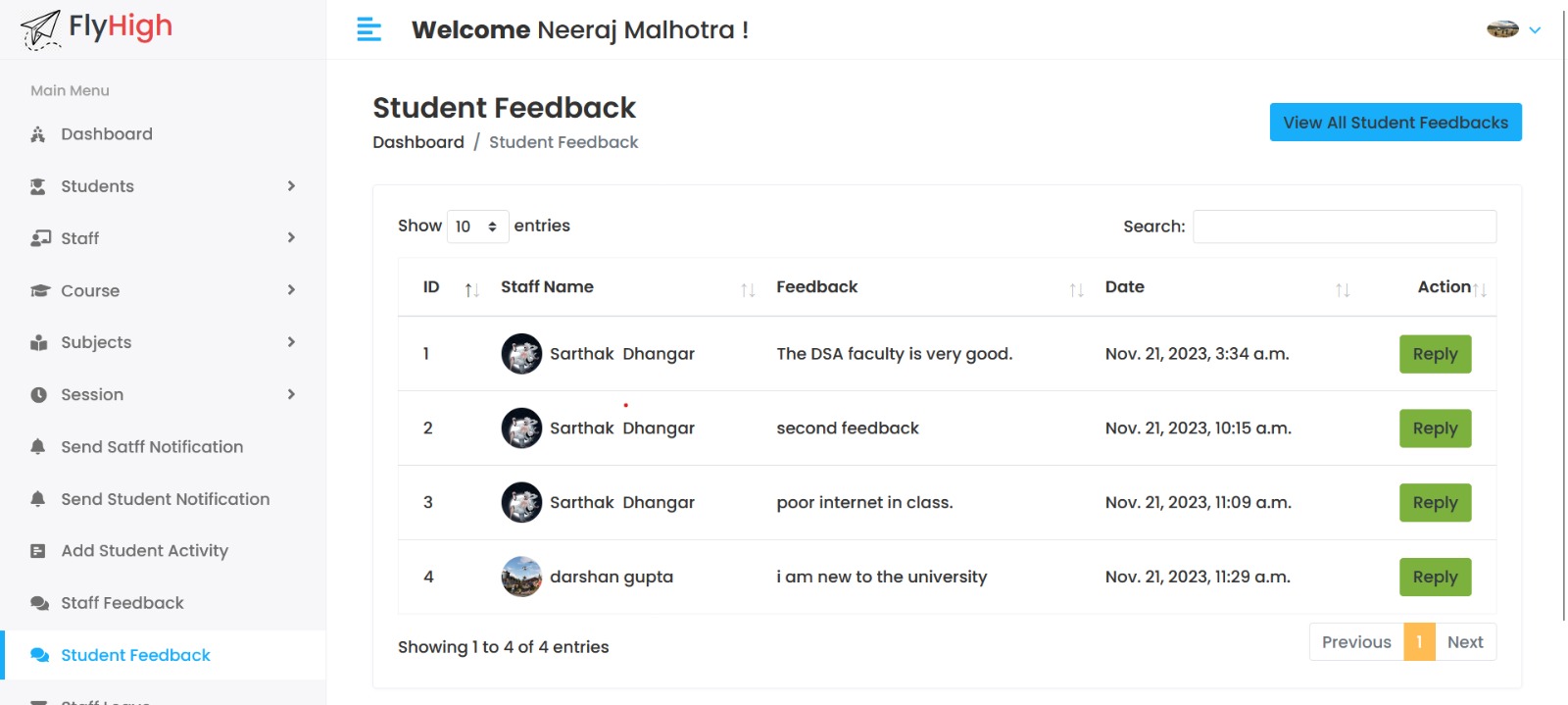
****

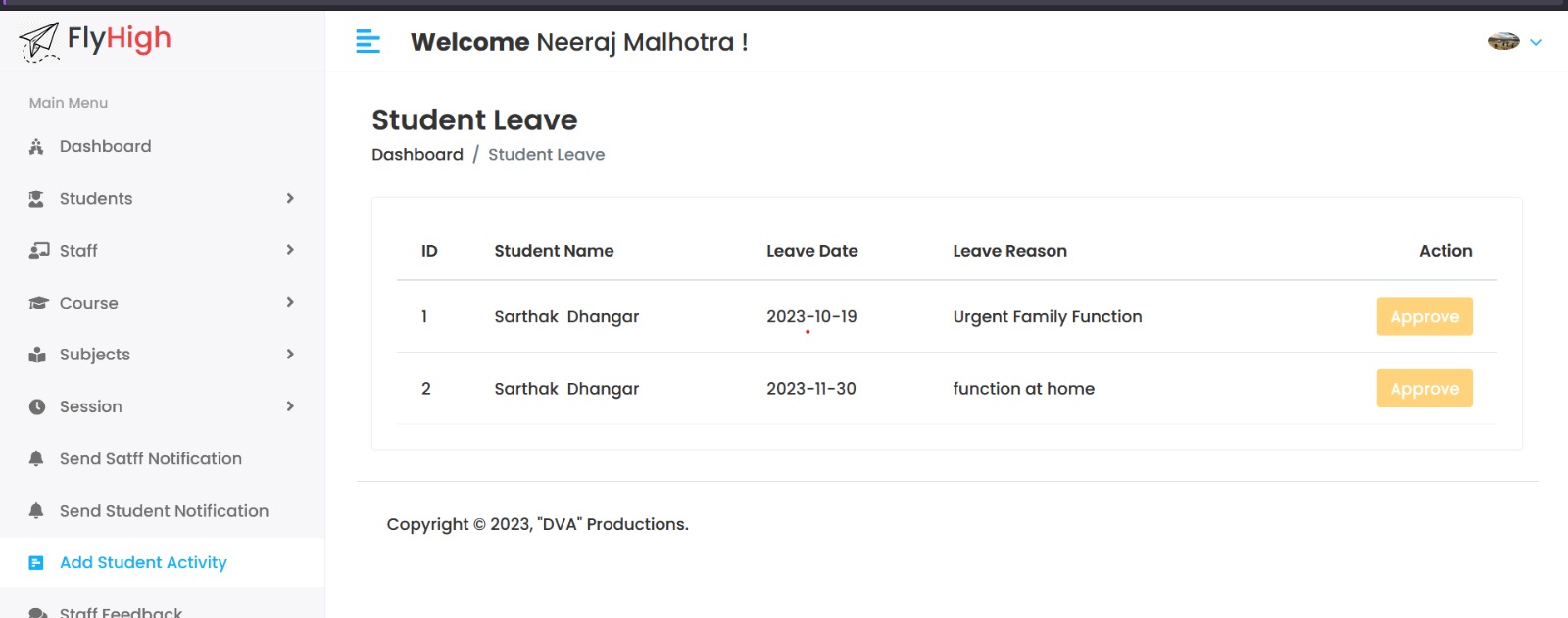
****

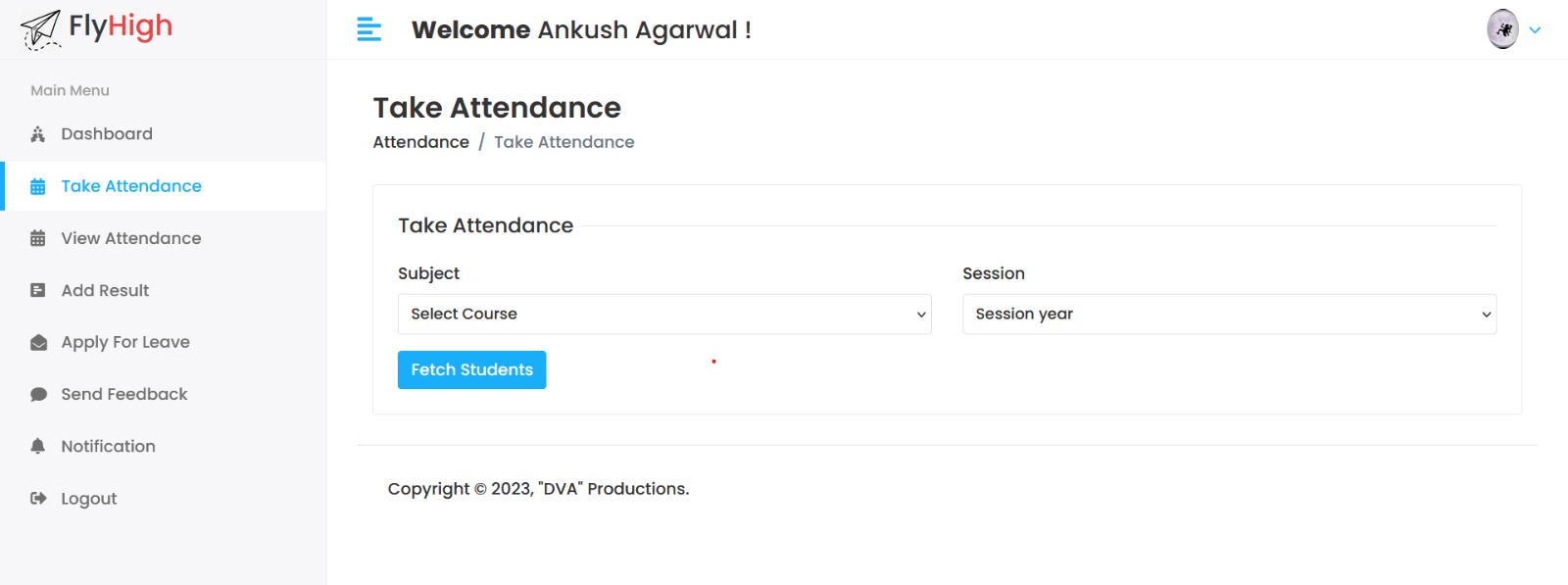
****

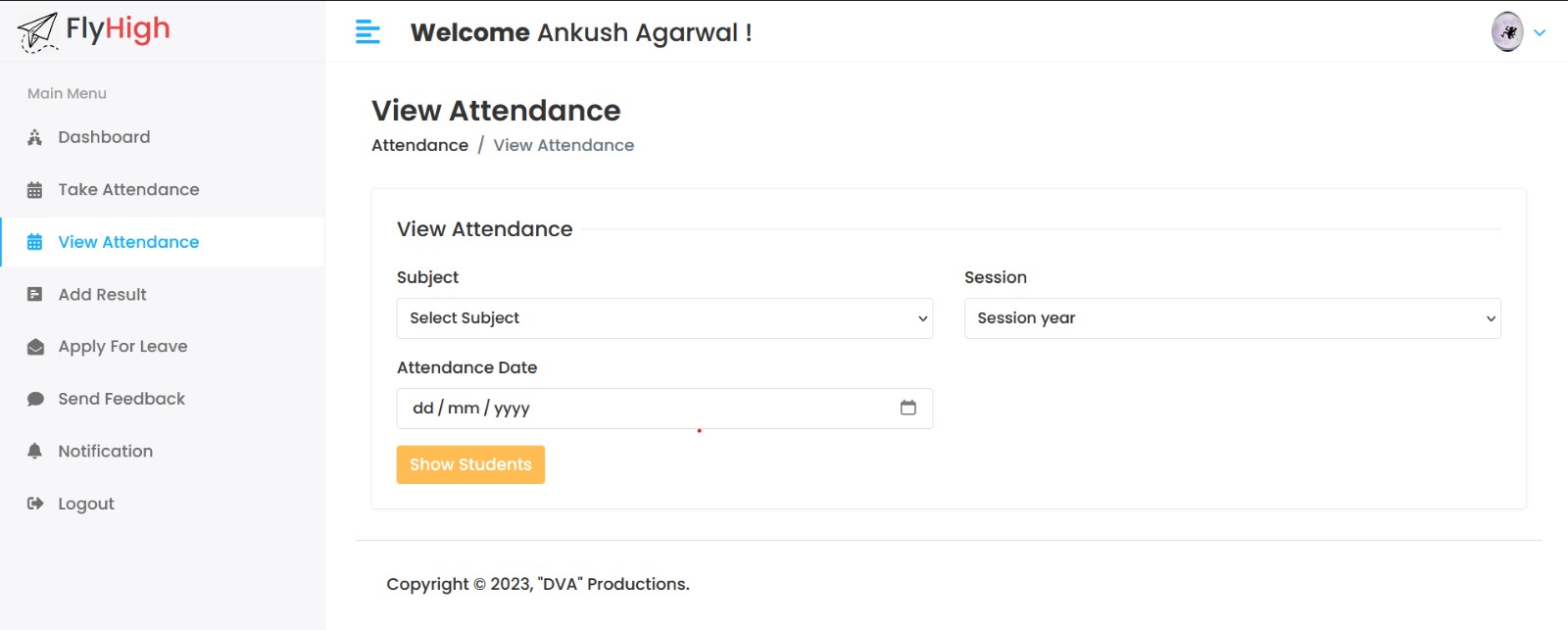
****

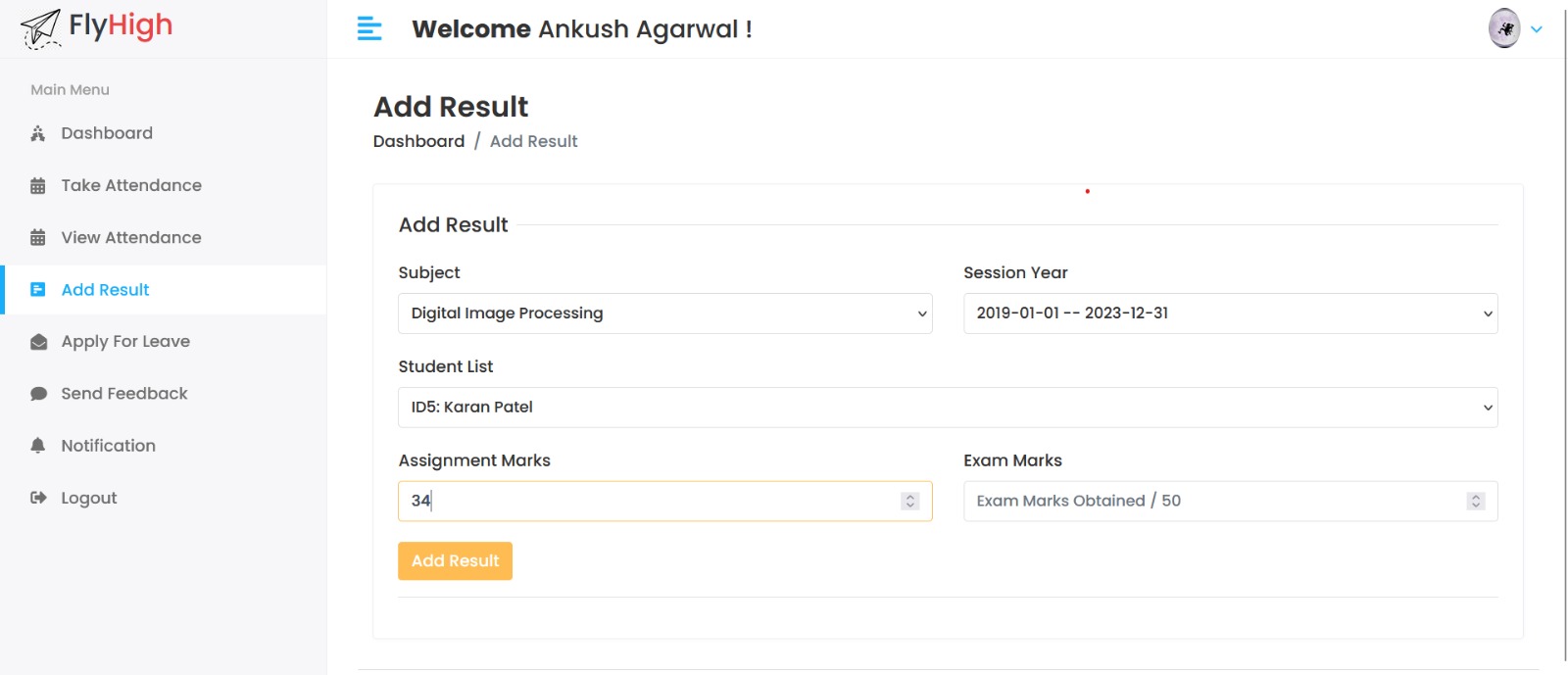
****

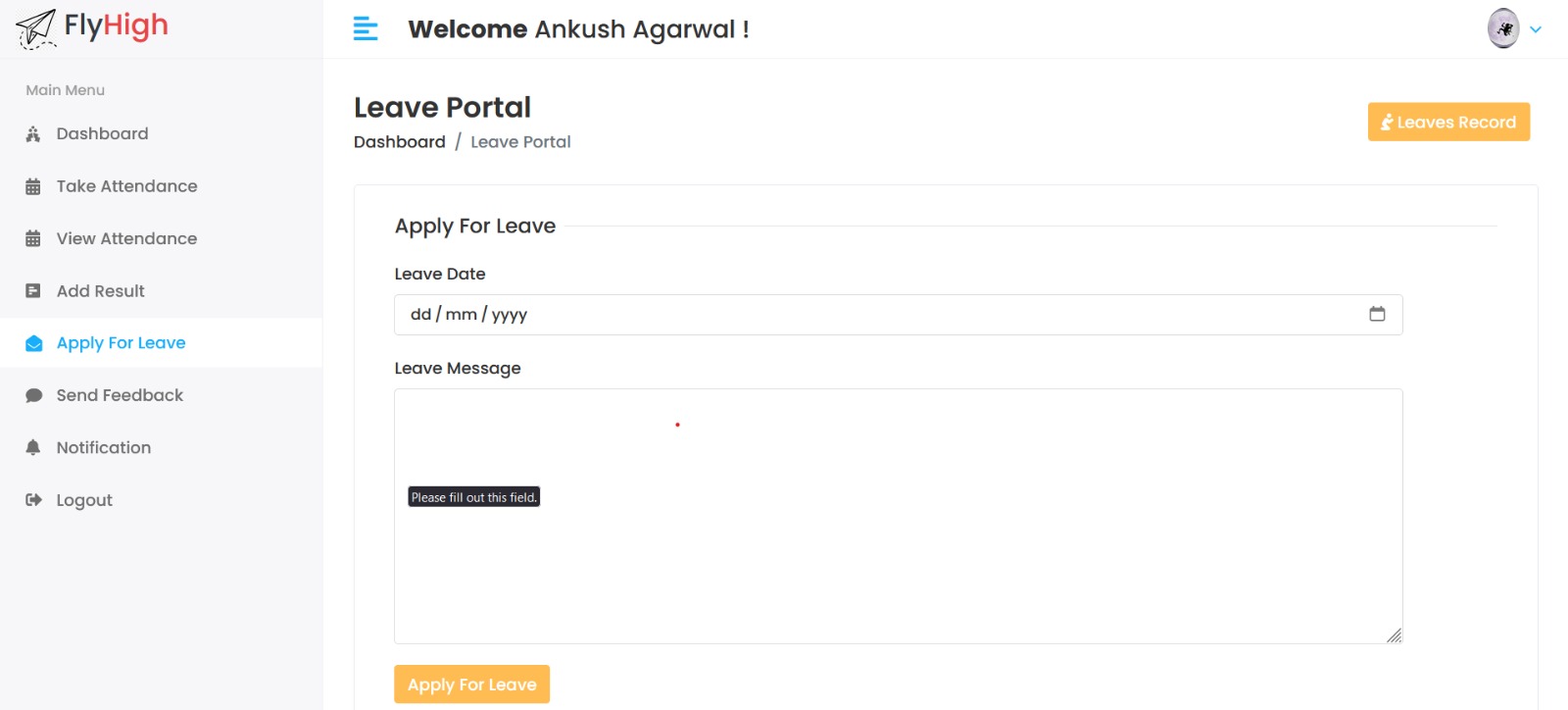
****

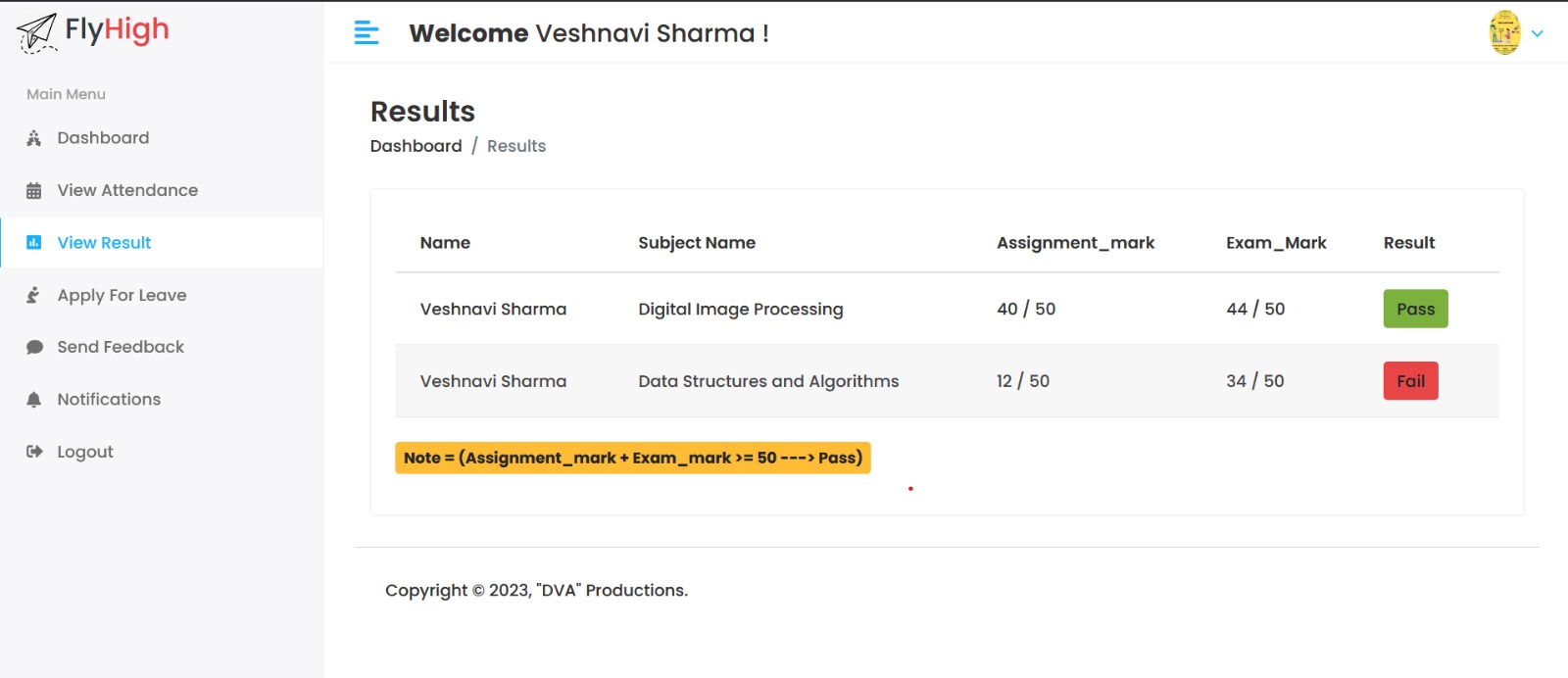
****

****

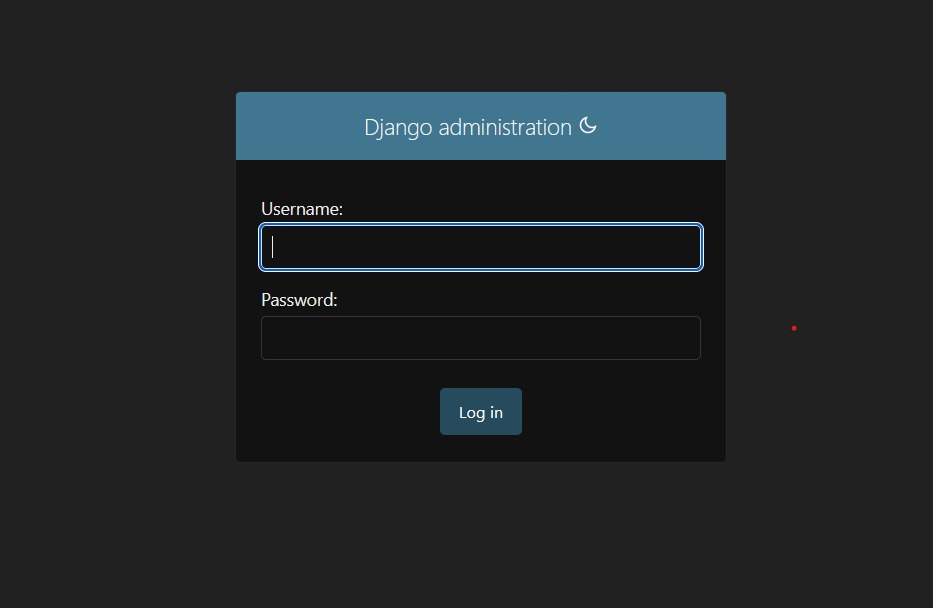
****

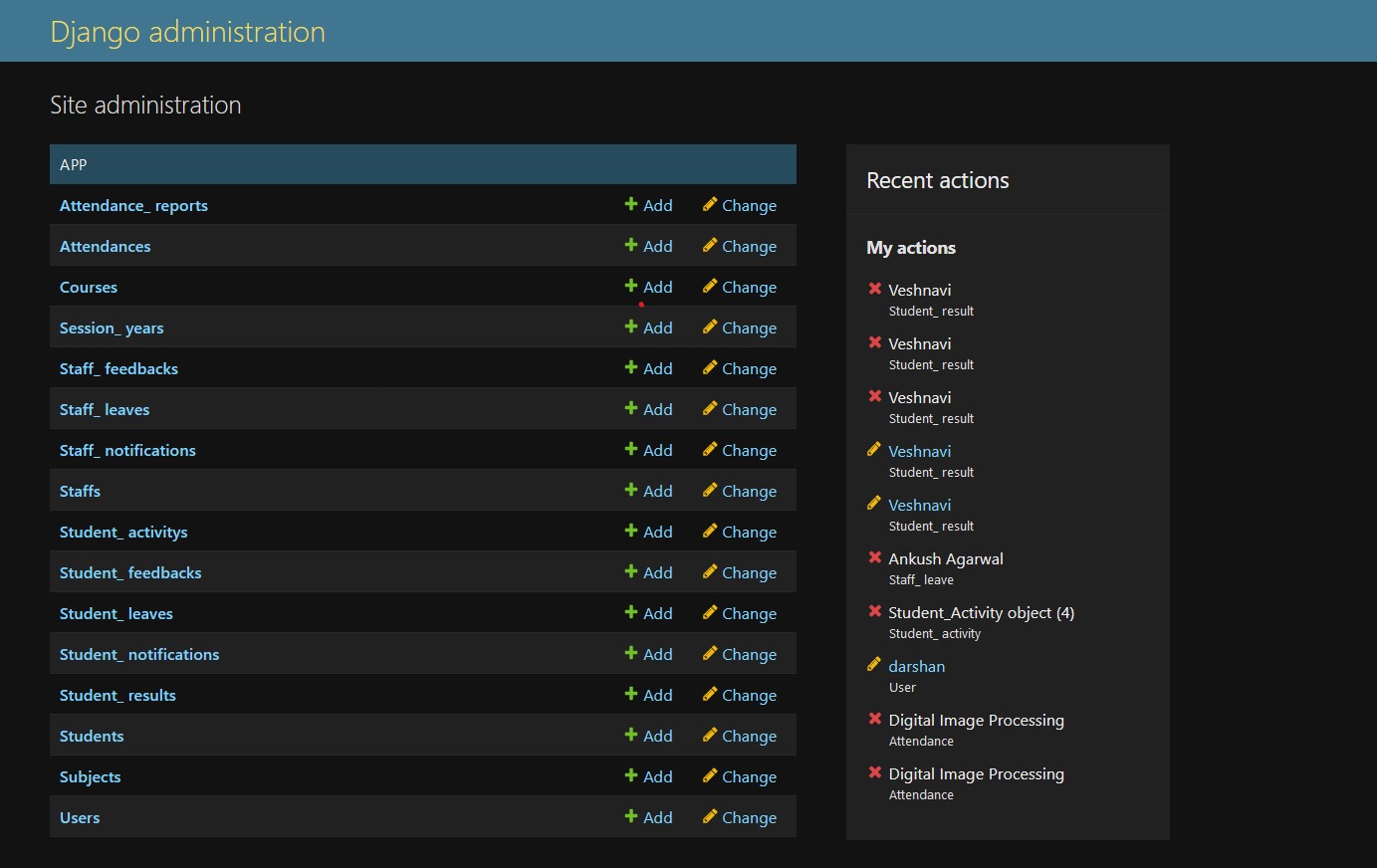
****

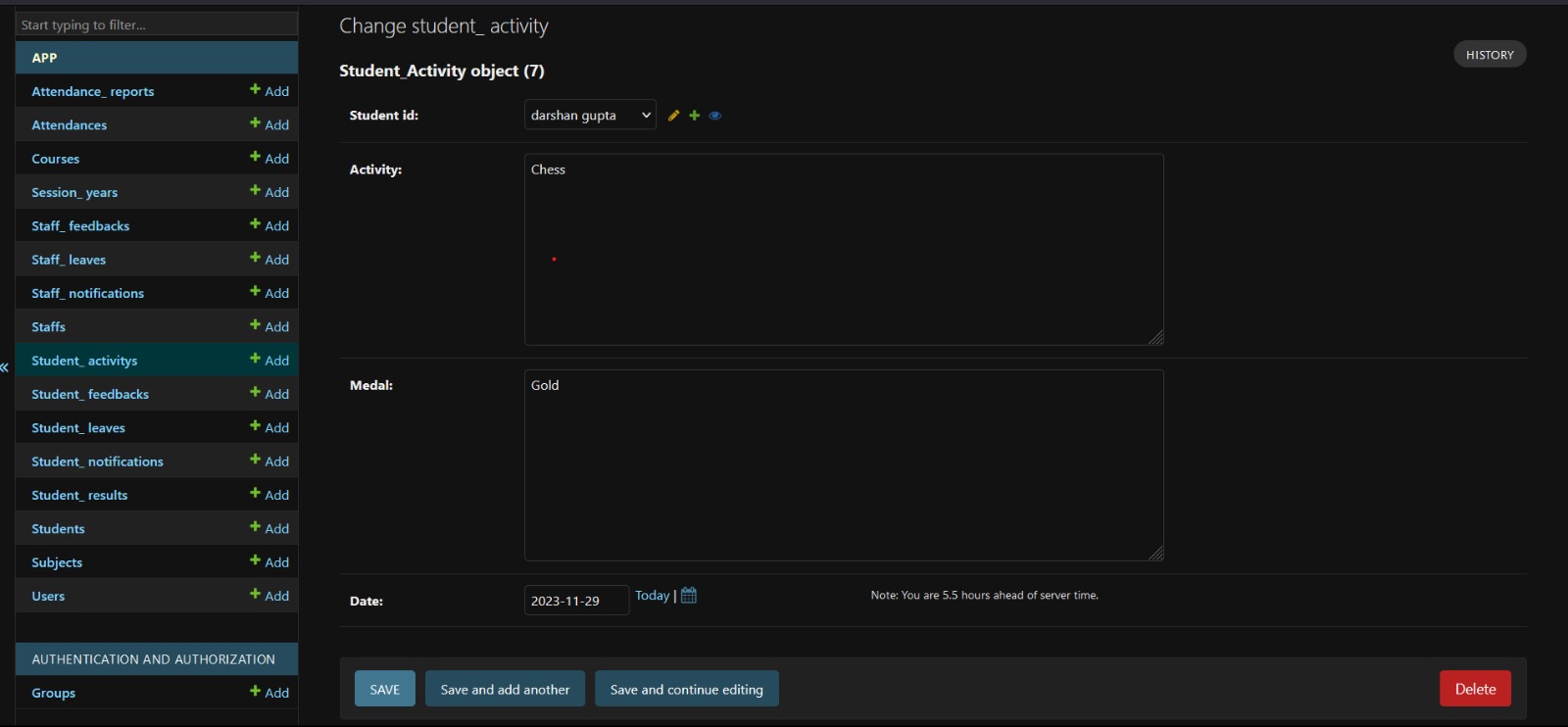
****

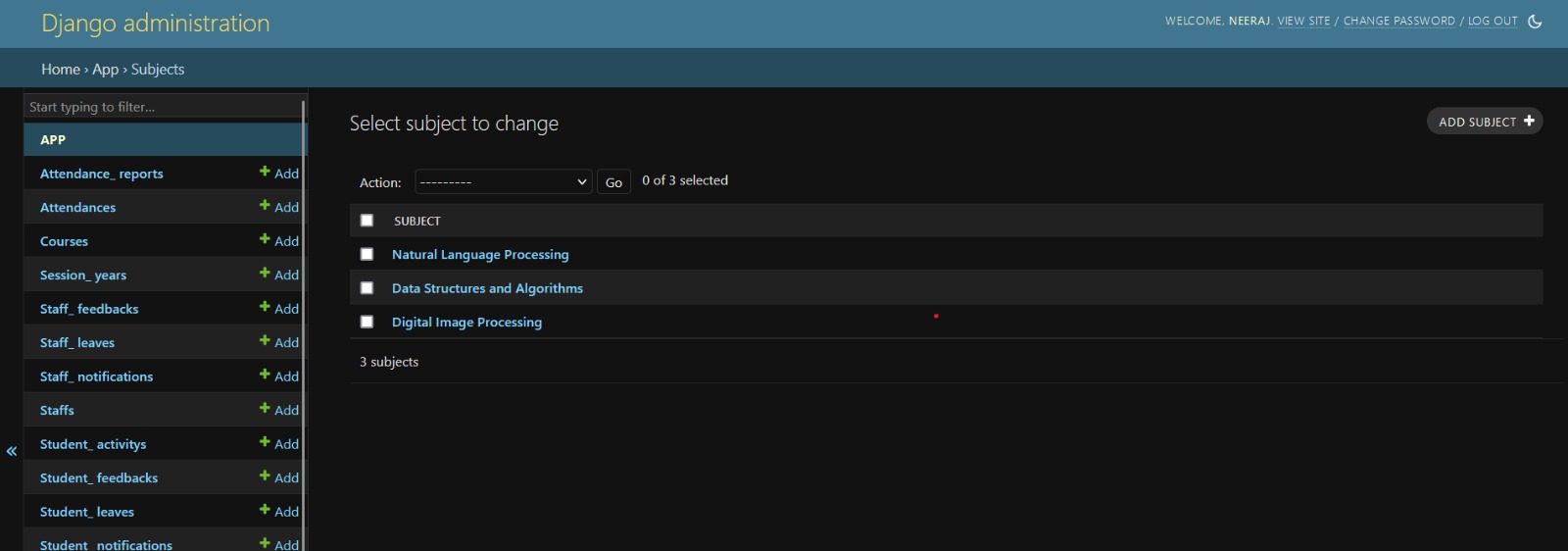
****

**BACKEND SCREENSHOTS**

****

****

****

****

CHAPTER-9

**MAINTENANCE**

The maintenance phase involves making changes to hardware, software, and documentation to support its operational effectiveness. It includes making changes to improve a system's performance, correct problems, enhance security, or address user requirements. To ensure modifications do not disrupt operations or degrade a system's performance or security, organizations should establish appropriate change management standards and procedures.

Routine changes are not as complex as major modifications and can usually be implemented in the normal course of business. Routine change controls should include procedures for requesting, evaluating, approving, testing, installing, and documenting website modifications. Maintaining accurate, up-to-date hardware and software inventories is a critical part of all change management processes. Management should carefully document all modifications to ensure accurate system inventories. Management should coordinate all technology related changes through an oversight committee and assign an appropriate party responsibility for administering software patch management programs. Quality assurance, security, audit, regulatory compliance, network, and end-user personnel should be appropriately included in change management processes. Risk and security review should be done whenever a system modification is implemented to ensure controls remain in place.

For maintenance of the website:

1. The database has to be updated regularly according to new available information.

2. Redundant and false information must be removed from the database.

3. Newer version of Django and Dbsqlite (Inbuilt) can be used for up gradation of website and to improve the overall performance of the system.

CHAPTER-10

**FUTURE SCOPE & FUTURE ENHANCEMENT**

**PROJECT NAME: FLY HIGH**

* **Integration of Artificial Intelligence (AI):**

1. Implement AI for predictive analytics to identify at-risk students and provide early interventions.

2. Develop AI-driven chatbots for student support and guidance.

3. Use machine learning algorithms to personalize learning paths based on individual student performance.

* **E-Learning Integration:**

1. Integrate e-learning platforms and resources directly into the SMS for a unified educational experience.

2. Enable features like virtual classrooms, online assessments, and collaborative learning tools

* **Mobile Applications and Accessibility:**

1. Develop mobile applications for student and teacher interfaces, allowing easy access to information from anywhere.
2. Ensure compatibility with various devices and platforms for seamless user experience.

CHAPTER-11

**CONCLUSION**

We have successfully implemented the site *'****Student Management System*’**. With the help of various links and tools, we have been able to provide a site which will be live soon and running on the web. We have been successful in our attempt to take care of the needs of both the user as well as the administrators. Finally we hope that this will go a long way in popularizing.

**BIBLIOGRAPHY**

1. [www.getbootstrap.com](http://www.getbootstrap.com)
2. [www.fontawesome.com](http://www.fontawesome.com)
3. [www.w3schools.com](http://www.w3schools.com)
4. [www.stackoverflow.com](http://www.stackoverflow.com)
5. Head First with Django
6. Learned HTML and CSS faster
7. Wikipedia.