

# Documentation: Student Management System

## models.py

### EducationalStage

- **name**: CharField that represents the name of the educational stage.
- **STAGE\_CHOICES**: Choices for the educational stage.
- **\_\_str\_\_**: Returns the name of the educational stage.

### Classroom

- **name**: CharField that represents the name of the classroom.
- **stage**: CharField that represents the stage of the classroom.
- **educational\_stage**: ForeignKey to EducationalStage, represents the educational stage associated with the classroom.
- **fee\_per\_student**: DecimalField that represents the fee per student for the classroom.
- **\_\_str\_\_**: Returns the name of the classroom.
- **total\_students**: Method that returns the total number of students in the classroom.
- **total\_fees\_due**: Method that returns the total fees due for the classroom.
- **total\_paid\_students**: Method that returns the total number of paid students in the classroom.
- **total\_unpaid\_students**: Method that returns the total number of unpaid students in the classroom.
- **remaining\_tuitions**: Method that returns the total remaining tuitions for the classroom.

### AcademicYear

- **year**: CharField that represents the academic year.
- **\_\_str\_\_**: Returns the academic year.

### Student

- **name**: CharField that represents the name of the student.
- **national\_number**: IntegerField that represents the national number of the student.
- **age**: IntegerField that represents the age of the student.
- **gender**: CharField that represents the gender of the student.
- **date\_of\_birth**: DateField that represents the date of birth of the student.

- **academic\_year**: ForeignKey to AcademicYear, represents the academic year associated with the student.
- **classroom**: ManyToManyField to Classroom, represents the classrooms associated with the student.
- **total\_payments**: DecimalField that represents the total payments made by the student.
- **\_\_str\_\_**: Returns the name of the student.
- **remaining\_dues**: Method that returns the remaining dues for the student.
- **total\_tuition**: Method that returns the total tuition for the student.

#### Expense

- **classroom**: ForeignKey to Classroom, represents the classroom associated with the expense.
- **expense\_type**: CharField that represents the type of expense.
- **amount**: DecimalField that represents the amount of the expense.
- **date**: DateField that represents the date of the expense.
- **total\_owed**: DecimalField that represents the total owed amount for the expense.
- **\_\_str\_\_**: Returns the type of expense.
- **save**: Method that overrides the default save method to calculate the total owed amount and update the classroom's fee\_per\_student.

#### Tuition

- **student**: ForeignKey to Student, represents the student associated with the tuition.
- **installment\_number**: IntegerField that represents the installment number.
- **amount**: DecimalField that represents the amount of the installment.
- **paid**: BooleanField that indicates if the installment has been paid.
- **receipt\_date**: DateField that represents the date of the installment.
- **save**: Method that overrides the default save method to update the student's total\_payments.

#### urls.py

- **app\_name**: Specifies the namespace for the URL patterns.
- **urlpatterns**: Contains the URL patterns for the student management system.

#### Views.py

- **home**: This view renders the home page and displays a list of students. It retrieves all the student objects from the database and passes them to the template to be displayed.

- **search\_student:** This view renders the search student page, which allows users to search for students based on certain criteria. It handles the search query submitted by the user and performs a database query to find matching student objects. The search results are then displayed on the page.
- **student\_list:** This view renders the student list page and displays a paginated list of students. It retrieves all the student objects from the database and uses pagination to display a subset of students on each page. The view also calculates and displays student statistics, such as the total number of students, total fees due, total paid students, and total unpaid students.
- **edit\_student:** This view renders the edit student form, which allows users to update student information. It handles the form submission and updates the corresponding student object in the database with the new data provided by the user.

**explanation of the function of each function in the views.py file:**

1. **delete\_student(request, student\_id):** This view is responsible for deleting a student object from the database. It retrieves the student object based on the **student\_id** parameter, and if the request method is POST, it deletes the student object and redirects to the home page. Otherwise, it renders the **delete\_student.html** template with the student object.
2. **add\_student(request):** This view handles the addition of a new student. If the request method is POST, it validates the student form data, saves the student object, and redirects to the add student page. If the method is GET, it renders the **add\_student.html** template with an empty student form.
3. **add\_expense(request, classroom\_id):** This view is responsible for adding expenses for a specific classroom. It retrieves the classroom object based on the **classroom\_id** parameter and the associated students. If the request method is POST, it creates an expense object for each student in the classroom with the provided expense details and redirects back to the same page. If the method is GET, it renders the **add\_expense.html** template with the classroom object.
4. **pay\_installment(request, pk):** This view handles the payment of an installment for a student. It retrieves the student object based on the **pk** parameter. If the request method is POST, it validates the tuition form data, saves the installment object with the student and user information, and redirects to the student detail page. If the method is GET, it renders the **pay\_installment.html** template with the student and installment form.
5. **add\_comment(request):** This view handles the addition of a comment. If the request method is POST, it saves the comment form data and redirects to the comment list page. If the method is GET, it renders the **add\_comment.html** template with an empty comment form.
6. **comment\_list(request):** This view retrieves all comments from the database and renders the **comment\_list.html** template with the comments.
7. **receipt(request, pk):** This view generates a receipt for a tuition installment. It retrieves the tuition object based on the **pk** parameter and calculates various totals and expenses related to the student and classroom. It renders the **receipt.html** template with the receipt details.

8. **student\_detail(request, pk)**: This view displays the details of a student. It retrieves the student object based on the **pk** parameter and calculates totals and expenses related to the student and classroom. It renders the **student\_detail.html** template with the student details.
9. **report(request)**: This view generates a report with various statistics and totals related to students, installments, genders, and classrooms. It retrieves data from the database, performs calculations, and renders the **report.html** template with the report details.