Documentation

This API provides endpoints to manage students, their marks, and various statistics related to their performance.

Endpoints

1. Get Students

- **Endpoint:** `/api/students`
- **Method:** `GET`
- **Description:** Retrieves a list of all students along with their details and marks.
- **Query:**

```sql
SELECT students.*, fields.id AS fieldId, fields.name AS
fieldName, marks.id AS markId, marks.subjectId, marks.marks,
marks.createdAt AS markCreatedAt, marks.updatedAt AS
markUpdatedAt
FROM students
LEFT JOIN fields ON students.fieldId = fields.id
LEFT JOIN marks ON students.id = marks.studentId
```

2. Get Student by ID

- **Endpoint:** `/api/students/:id`
- **Method:** `GET`
- **Description:** Retrieves details of a specific student by their ID.
- **Query:**

```sql
SELECT students.*, fields.id AS fieldId, fields.name AS
fieldName, marks.id AS markId, marks.subjectId, marks.marks,
marks.createdAt AS markCreatedAt, marks.updatedAt AS
markUpdatedAt
FROM students
LEFT JOIN fields ON students.fieldId = fields.id
LEFT JOIN marks ON students.id = marks.studentId
WHERE students.id = ?
```

3. Update Student

- **Endpoint:** `/api/students/:id`
- **Method:** `PUT`
- **Description:** Updates details of a specific student by their ID.

- **Query:**

```sql
UPDATE students
SET username = ?, enrollmentYear = ?, fieldId = ?
WHERE id = ?
```

4. Delete Student

- **Endpoint:** `/api/students/:id`
- **Method:** `DELETE`
- **Description:** Deletes a specific student by their ID.
- **Query:**

```sql
DELETE FROM students WHERE id = ?
```

5. Add or Update Mark

- **Endpoint:** `/api/marks`
- **Method:** `POST`
- **Description:** Adds or updates marks for a student in a specific subject.
- **Query:** (Handled by Sequelize ORM)

6. Get Average Total Marks

- **Endpoint:** `/api/statistics/average-total-marks`
- **Method:** `GET`
- **Description:** Calculates the average total marks for each field and each subject.
- **Query:**

```sql
-- For fields
SELECT
  fields.name AS fieldName,
  AVG(marks.marks) AS averageMarks
FROM students
JOIN fields ON students.fieldId = fields.id
JOIN marks ON students.id = marks.studentId
GROUP BY fields.name

-- For subjects within fields
SELECT
  fields.name AS fieldName,
  subjects.name AS subjectName,
  AVG(marks.marks) AS averageMarks
FROM students
JOIN fields ON students.fieldId = fields.id
JOIN marks ON students.id = marks.studentId
```

```
JOIN subjects ON marks.subjectId = subjects.id
GROUP BY fields.name, subjects.name
```

7. Get Subject-Wise Highest Marks

- **Endpoint:** `/api/statistics/subject-wise-highest-marks`
- **Method:** `GET`
- **Description:** Retrieves the highest marks scored in each subject across different fields.
- **Query:**

```
SELECT
  fields.name AS fieldName,
  subjects.name AS subjectName,
  MAX(marks.marks) AS highestMarks
FROM students
JOIN fields ON students.fieldId = fields.id
JOIN marks ON students.id = marks.studentId
JOIN subjects ON marks.subjectId = subjects.id
GROUP BY fields.name, subjects.name
```

8. Get Top Students

- **Endpoint:** `/api/statistics/top-students`
- **Method:** `GET`
- **Description:** Retrieves the top-performing students in each field based on total marks.
- **Query:**

```
SELECT fields.name AS fieldName, students.id AS studentId,
students.username AS studentName, SUM(marks.marks) AS totalMarks
FROM students
JOIN fields ON students.fieldId = fields.id
JOIN marks ON students.id = marks.studentId
GROUP BY fields.name, students.id, students.username
ORDER BY fields.name, totalMarks DESC LIMIT 3
```

9. Get Subject Pass Rate

- **Endpoint:** `/api/statistics/subject-pass-rate`
- **Method:** `GET`
- **Description:** Calculates the pass rate for each subject within each field.
- **Query:**

```
SELECT
  f.name AS fieldName,
  s.name AS subjectName,
  AVG(CASE WHEN m.marks >= 60 THEN 1 ELSE 0 END) * 100 AS
passRate
FROM
  Fields f
JOIN
  Students stu ON f.id = stu.fieldId
JOIN
  Marks m ON stu.id = m.studentId
JOIN
  Subjects s ON m.subjectId = s.id
GROUP BY
  f.name, s.name
```

10. Get Students Count by Field

- **Endpoint:** `/api/statistics/students-count-by-field`
- **Method:** `GET`
- **Description:** Retrieves the count of students in each field.
- **Query:**

```
SELECT
  fields.name AS fieldName,
  COUNT(*) AS studentCount
FROM students
JOIN fields ON students.fieldId = fields.id
GROUP BY fields.name
```