**API Design Document**

**API Endpoints**

**Students**

**1. Create a Student**

* **Endpoint**: POST /students
* **Description**: Creates a new student.
* **Request Body**:

.

.

{

"name": "string",

"email": "string"

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**2. Get All Students**

* **Endpoint**: GET /students
* **Description**: Retrieves all students with pagination.
* **Query Parameters**:
  + page (optional): Page number (default is 1)
  + limit (optional): Number of records per page (default is 10)
* **Response**:

.

.

{

"totalStudents": "number",

"totalPages": "number",

"currentPage": "number",

"students": [

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

]

}

**3. Get a Student by ID**

* **Endpoint**: GET /students/:id
* **Description**: Retrieves a student by their ID.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**4. Update a Student**

* **Endpoint**: PUT /students/:id
* **Description**: Updates a student's information.
* **Request Body**:

.

.

{

"name": "string",

"email": "string"

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**5. Delete a Student**

* **Endpoint**: DELETE /students/:id
* **Description**: Deletes a student.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**6. Get a Student's Courses for a Specific Class**

* **Endpoint**: GET /students/:id/classes/:classId/courses
* **Description**: Retrieves a student's enrolled courses for a specific class.
* **Response**:

.

.

[

{

"\_id": "string",

"name": "string"

}

]

**7. Enroll a Student in a Class**

* **Endpoint**: POST /students/:id/classes/:classId/enroll
* **Description**: Enrolls a student in a class and its associated courses.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": ["string"],

"courses": ["string"]

}

**8. Enroll a Student in a Course**

* **Endpoint**: POST /students/:id/classes/:classId/courses/:courseId/enroll
* **Description**: Enrolls a student in a specific course within a class.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": ["string"],

"courses": ["string"]

}

**Teachers**

**1. Create a Teacher**

* **Endpoint**: POST /teachers
* **Description**: Creates a new teacher.
* **Request Body**:

.

.

{

"name": "string",

"email": "string"

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**2. Get All Teachers**

* **Endpoint**: GET /teachers
* **Description**: Retrieves all teachers with pagination.
* **Query Parameters**:
  + page (optional): Page number (default is 1)
  + limit (optional): Number of records per page (default is 10)
* **Response**:

.

.

{

"totalTeachers": "number",

"totalPages": "number",

"currentPage": "number",

"teachers": [

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

]

}

**3. Get a Teacher by ID**

* **Endpoint**: GET /teachers/:id
* **Description**: Retrieves a teacher by their ID.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**4. Update a Teacher**

* **Endpoint**: PUT /teachers/:id
* **Description**: Updates a teacher's information.
* **Request Body**:

.

.

{

"name": "string",

"email": "string"

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**5. Delete a Teacher**

* **Endpoint**: DELETE /teachers/:id
* **Description**: Deletes a teacher.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"email": "string",

"classes": [],

"courses": []

}

**Courses**

**1. Create a Course**

* **Endpoint**: POST /courses
* **Description**: Creates a new course.
* **Request Body**:

.

.

{

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

**2. Get All Courses**

* **Endpoint**: GET /courses
* **Description**: Retrieves all courses with pagination.
* **Query Parameters**:
  + page (optional): Page number (default is 1)
  + limit (optional): Number of records per page (default is 10)
* **Response**:

.

.

{

"totalCourses": "number",

"totalPages": "number",

"currentPage": "number",

"courses": [

{

"\_id": "string",

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

]

}

**3. Get a Course by ID**

* **Endpoint**: GET /courses/:id
* **Description**: Retrieves a course by its ID.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

**4. Update a Course**

* **Endpoint**: PUT /courses/:id
* **Description**: Updates a course's information.
* **Request Body**:

.

.

{

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

**5. Delete a Course**

* **Endpoint**: DELETE /courses/:id
* **Description**: Deletes a course.
* **Response**:

.

.

{

"\_id": "string",

"name": "string",

"teachers": ["string"],

"classes": ["string"]

}

**Classes**

**1. Create a Class**

* **Endpoint**: POST /classes
* **Description**: Creates a new class.
* **Request Body**:

.

.

{

"name": "string"

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string"

}

**2. Get All Classes**

* **Endpoint**: GET /classes
* **Description**: Retrieves all classes with pagination.
* **Query Parameters**:
  + page (optional): Page number (default is 1)
  + limit (optional): Number of records per page (default is 10)
* **Response**:

.

.

{

"totalClasses": "number",

"totalPages": "number",

"currentPage": "number",

"classes": [

{

"\_id": "string",

"name": "string"

}

]

}

**3. Get a Class by ID**

* **Endpoint**: GET /classes/:id
* **Description**: Retrieves a class by its ID.
* **Response**:

.

.

{

"\_id": "string",

"name": "string"

}

**4. Update a Class**

* **Endpoint**: PUT /classes/:id
* **Description**: Updates a class's information.
* **Request Body**:

.

.

{

"name": "string"

}

* **Response**:

.

.

{

"\_id": "string",

"name": "string"

}

**5. Delete a Class**

* **Endpoint**: DELETE /classes/:id
* **Description**: Deletes a class.
* **Response**:

.

.

{

"\_id": "string",

"name": "string"

}

**Marks**

**1. Create a Mark**

* **Endpoint**: POST /marks
* **Description**: Creates a new mark entry.
* **Request Body**:

.

.

{

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

* **Response**:

.

.

{

"\_id": "string",

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

**2. Get All Marks**

* **Endpoint**: GET /marks
* **Description**: Retrieves all marks.
* **Response**:

.

.

[

{

"\_id": "string",

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

]

**3. Get a Mark by ID**

* **Endpoint**: GET /marks/:id
* **Description**: Retrieves a mark by its ID.
* **Response**:

.

.

{

"\_id": "string",

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

**4. Add a Mark**

* **Endpoint**: POST /marks/add
* **Description**: Adds a new mark entry for a student in a specific course within a class.
* **Request Body**:

.

.

{

"studentId": "string",

"courseId": "string",

"classId": "string",

"teacherId": "string",

"marks": "number"

}

* **Response**:

.

.

{

"\_id": "string",

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

**5. Add Multiple Marks**

* **Endpoint**: POST /marks/add-multiple
* **Description**: Adds multiple mark entries for a student in various courses within a class.
* **Request Body**:

.

.

{

"studentId": "string",

"classId": "string",

"marks": [

{

"courseId": "string",

"teacherId": "string",

"marks": "number"

}

]

}

* **Response**:

.

.

[

{

"\_id": "string",

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

]

**6. Get Marks by Course**

* **Endpoint**: GET /marks/course/:courseId
* **Description**: Retrieves all marks for a specific course.
* **Response**:

.

.

[

{

"\_id": "string",

"student": "string",

"course": "string",

"class": "string",

"teacher": "string",

"marks": "number"

}

]

**ERD Diagram**

