

README :

Java Backend Project with MongoDB (Student Management System)

By Debajyoti Banerjee
(Developed for local host deployment)

Project Overview

This document serves as the README file for the "Student Management System" (SMS) backend project, a Java-based application integrated with MongoDB. This is the initial V1.0 build, primarily designed for local development and testing.

Tech Stack

- **Java (Maven Project) :** Core development language for the backend application.
- **MongoDB:** Utilized as the NoSQL, JSON document-based database. Interactions are primarily managed via the Java driver, with mongosh (MongoDB Shell) used for direct database inspection and administrative tasks.
 - **Note:** For full accessibility during development and testing, it is recommended to run CMD Prompt as an administrator when invoking mongosh.
 -

Development Environment & Resources

- JDK Version: 24.0.1
- IDE: Eclipse IDE 2023-12 (Other compatible IDEs can also be used).
- MongoDB Version: 8.0.10 (gitVersion: 9d03076bb2d5147d5b6fe381c7118b0b0478b682)
- Documentation: MongoDB Atlas official documentation.

This initial build of the SMS demonstrates core integration with MongoDB. While foundational features are implemented, some advanced functionalities are slated for future releases. Potential minor bugs related to legacy JRE versions within the Eclipse IDE may exist.

Getting Started:

1. Install MongoDB Atlas:-

a. Complete the necessary steps as follows:

- i. Download the MongoDB.exe file via this link given below:
https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-8.0.10-signed.msi
- ii. Download the MongoDB Shell.exe file via this link given below:
<https://downloads.mongodb.com/compass/mongosh-2.5.2-win32-x64.zip>
- iii. After downloading the mongodb.exe file follow steps as shown by the windows installation wizard and select **Complete Setup** and click **Next**, it will also ask if you want to install the MongoDB GUI Compass application and after installation is completed click on finish. You are good to go.
- iv. **Environment Path-Setup:** (By Default Path is like this located in C Drive inside program files locate the MongoDB folder) (**for reference purpose only**):
C:\Program Files\MongoDB\Server\8.0\bin

- v. Open windows setting-> type->"**Environment Variables**" under Environment variable Dialogue box(You will see advanced tab column) **select->"Environment variables"** then under System Variables **select->Path->Edit->New->**add the path provided above in **step iv** then **Hit Ok->**then again click-> **OK**

2. Java JDK installation:

a. Complete the necessary steps as follows:

- i. Download via given download link:
https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.msi
- ii. After downloading the file, follow the installation steps instructed by the Windows Installation wizard till the end then Hit **Finish** button.
- iii. Follow the path setup instruction provided for the MongoDB Bin will similar for the Path Setup locate and copy the path of **Java/jdk24.0.1/bin** folder address to the environment setup and then click ok and close the dialogue box same as incase of MongoDB path setup steps.

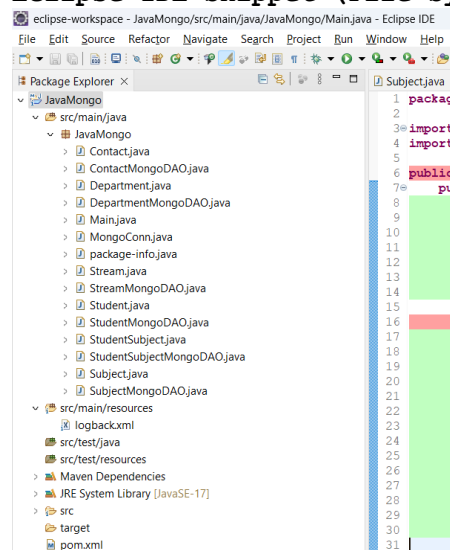
b. IDE installation: follow the Eclipse IDE download link provided:

- i. IDE Download link:
https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2025-06/R/eclipse-jee-2025-06-R-win32-x86_64.zip
- ii. Alternatively, any compatible IDE (e.g., IntelliJ IDEA, VS Code with Java extensions) can be used.

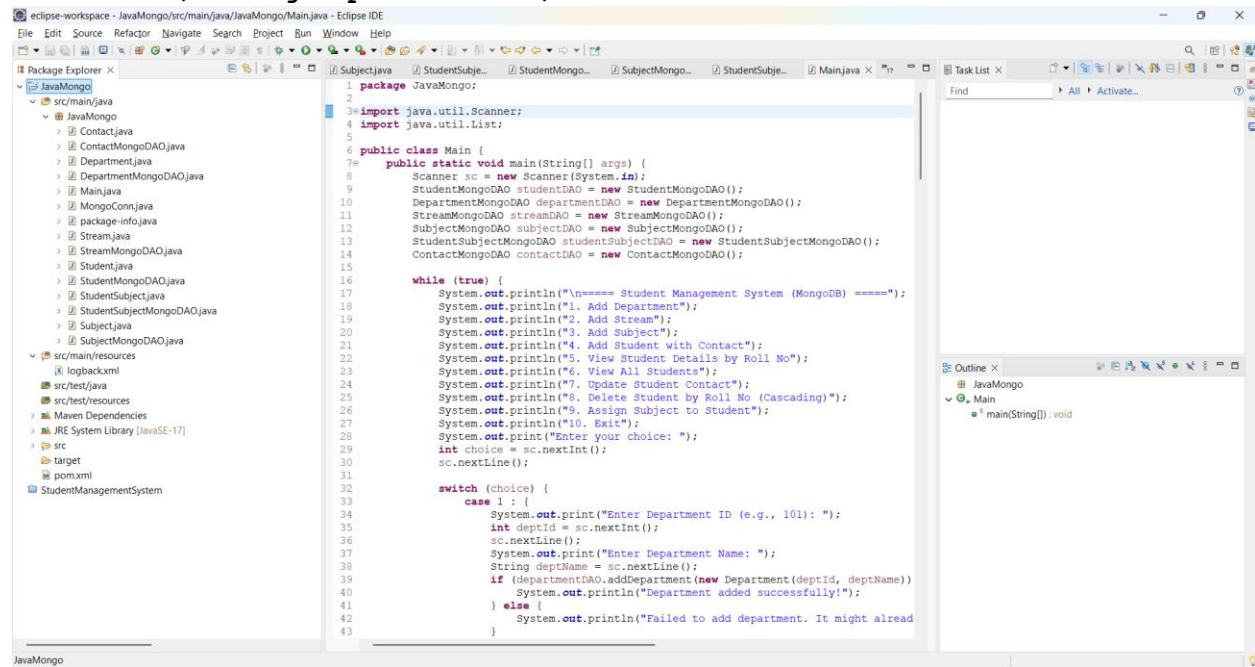
Project Structure and Testing

Below are snippets illustrating the project's file structure within the IDE and sample output from testing. White-box testing, following standard SDLC practices, was performed to validate the implemented features.

Eclipse IDE Snippet (File System Reference):

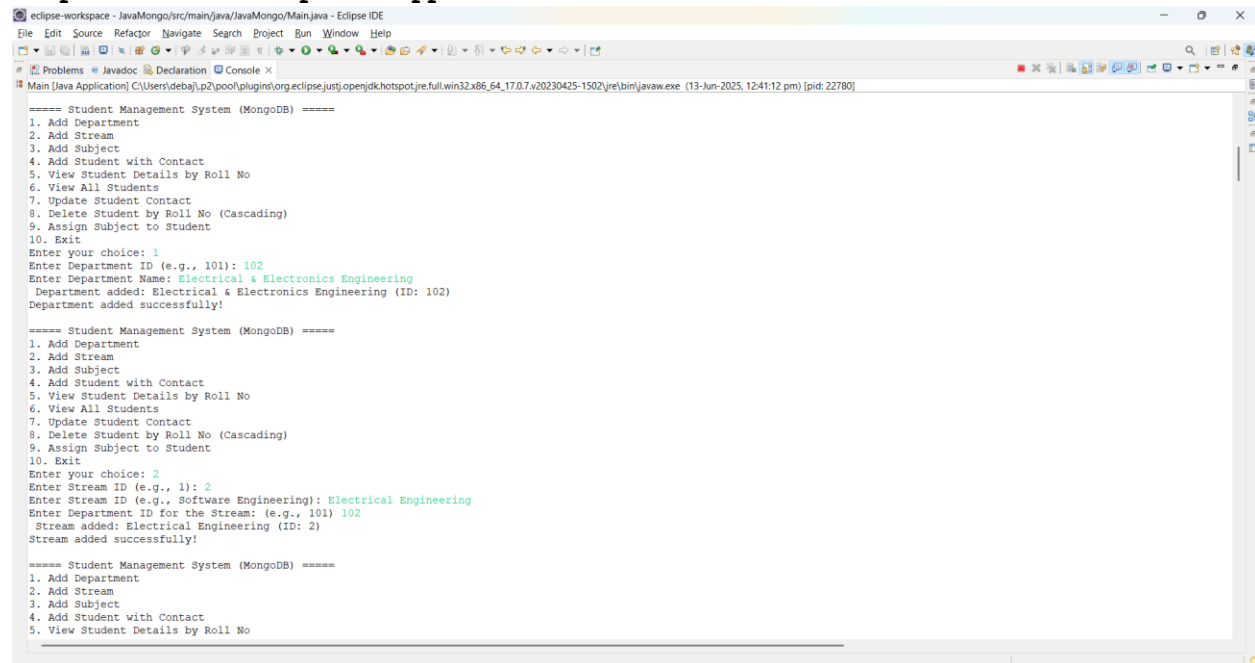


Here are the snippets: Main class (Testing Implementation):



```
1 package JavaMongo;
2
3 import java.util.Scanner;
4 import java.util.List;
5
6 public class Main {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         StudentMongoDAO studentDAO = new StudentMongoDAO();
10        DepartmentMongoDAO departmentDAO = new DepartmentMongoDAO();
11        StreamMongoDAO streamDAO = new StreamMongoDAO();
12        SubjectMongoDAO subjectDAO = new SubjectMongoDAO();
13        StudentSubjectMongoDAO studentSubjectDAO = new StudentSubjectMongoDAO();
14        ContactMongoDAO contactDAO = new ContactMongoDAO();
15
16        while (true) {
17            System.out.println("\n===== Student Management System (MongoDB) =====");
18            System.out.println("1. Add Department");
19            System.out.println("2. Add Stream");
20            System.out.println("3. Add Subject");
21            System.out.println("4. Add Student with Contact");
22            System.out.println("5. View Student Details by Roll No");
23            System.out.println("6. View All Students");
24            System.out.println("7. Update Student Contact");
25            System.out.println("8. Delete Student by Roll No (Cascading)");
26            System.out.println("9. Assign Subject to Student");
27            System.out.println("10. Exit");
28            System.out.print("Enter your choice: ");
29            int choice = sc.nextInt();
30            sc.nextLine();
31
32            switch (choice) {
33                case 1: {
34                    System.out.print("Enter Department ID (e.g., 101): ");
35                    int deptId = sc.nextInt();
36                    sc.nextLine();
37                    System.out.print("Enter Department Name: ");
38                    String deptName = sc.nextLine();
39                    if (departmentDAO.addDepartment(new Department(deptId, deptName))
40                        System.out.println("Department added successfully!");
41                    } else {
42                        System.out.println("Failed to add department. It might already exist.");
43                    }
44                }
45            }
46        }
47    }
48 }
```

Sample Console Output Snippet:



```
===== Student Management System (MongoDB) =====
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 1
Enter Department ID (e.g., 101): 102
Enter Department Name: Electrical & Electronics Engineering
Department added: Electrical & Electronics Engineering (ID: 102)
Department added successfully!

===== Student Management System (MongoDB) =====
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 2
Enter Stream ID (e.g., 1): 2
Enter Department ID for the Stream: (e.g., 101) 102
Stream added: Electrical Engineering (ID: 2)
Stream added successfully!

===== Student Management System (MongoDB) =====
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
```

```
MongoDB connection established to database: Students_db

===== Student Management System (MongoDB) =====
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 10
Exiting application. Goodbye!
```

Actual Output texts copied and pasted from the output display area "console" in eclipse IDE:

```
MongoDB connection established to database: Students_db

===== Student Management System (MongoDB) =====
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 1
Enter Department ID (e.g., 101): 102
Enter Department Name: Electrical & Electronics Engineering
  Department added: Electrical & Electronics Engineering (ID: 102)
Department added successfully!

===== Student Management System (MongoDB) =====
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 2
Enter Stream ID (e.g., 1): 2
Enter Stream ID (e.g., Software Engineering): Electrical Engineering
Enter Department ID for the Stream: (e.g., 101) 102
  Stream added: Electrical Engineering (ID: 2)
Stream added successfully!
```

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 3

Enter Subject ID (e.g., 201): 202

Enter Subject Name (e.g., Data Structures): RDBMS

Subject added: RDBMS (ID: 202)

Subject added successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 4

Enter Student Roll No (e.g., STU001): ST002

Enter Student Name (e.g., Alice Smith): Punit Jaiswal

Enter Semester (e.g., 1,2,3): 1

Enter Year of Admission (e.g., 2023): 2023

Enter Expected Passing Year (e.g., 2027): 2027

Enter Stream ID (e.g., 1 (Must match the stream ID you added earlier)): 1

Stream exists for ID: 1

Student added: Punit Jaiswal (Roll No: ST002)

Enter Contact Reg No (for this student) (e.g., CON001): CON002

Enter Phone (e.g., 1234567890): 9813448134

Enter Email (e.g., alice.s@example.com): punit@example.com

Contact added for Roll No: ST002

Student and contact added successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 9

Enter Student Roll No (e.g., STU001): STU002

Enter Subject ID to assign (e.g., 201): 201

Error: Student with Roll No STU002 does not exist.

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 9

Enter Student Roll No (e.g., STU001): ST002

Enter Subject ID to assign (e.g., 201): 202

Subject exists for ID: 202

Association does NOT exist for student ST002 and subject 202.

Assigned Subject ID 202 to Student Roll No: ST002

Subject assigned to student successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 3

Enter Subject ID (e.g., 201): 301

Enter Subject Name (e.g., Data Structures): Fundamental Electrical & Electronics

Subject added: Fundamental Electrical & Electronics (ID: 301)

Subject added successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 4

Enter Student Roll No (e.g., STU001): ETU001

Enter Student Name (e.g., Alice Smith): Harpreet Sukkhi

Enter Semester (e.g., 1,2,3): 1

Enter Year of Admission (e.g., 2023): 2023

Enter Expected Passing Year (e.g., 2027): 2027

Enter Stream ID (e.g., 1 (Must match the stream ID you added earlier): 2

Stream exists for ID: 2

Student added: Harpreet Sukkhi (Roll No: ETU001)
Enter Contact Reg No (for this student) (e.g., CON001): CON003
Enter Phone (e.g., 1234567890): 4564566789
Enter Email (e.g., alice.s@example.com): harpreet@example.com
Contact added for Roll No: ETU001
Student and contact added successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 9
Enter Student Roll No (e.g., STU001): ETU001
Enter Subject ID to assign (e.g., 201): 301
Subject exists for ID: 301
Association does NOT exist for student ETU001 and subject 301.
Assigned Subject ID 301 to Student Roll No: ETU001
Subject assigned to student successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 4
Enter Student Roll No (e.g., STU001): ETU002
Enter Student Name (e.g., Alice Smith): Gurmeet Singh
Enter Semester (e.g., 1,2,3): 1
Enter Year of Admission (e.g., 2023): 2023
Enter Expected Passing Year (e.g., 2027): 2027
Enter Stream ID (e.g., 1 (Must match the stream ID you added earlier)): 2
Stream exists for ID: 2
Student added: Gurmeet Singh (Roll No: ETU002)
Enter Contact Reg No (for this student) (e.g., CON001): CON004
Enter Phone (e.g., 1234567890): 2345123451
Enter Email (e.g., alice.s@example.com): gurmeet@example.com
Contact added for Roll No: ETU002
Student and contact added successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact

```
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 9
Enter Student Roll No (e.g., STU001): ETU002
Enter Subject ID to assign (e.g., 201): 301
  Subject exists for ID: 301
  Association does NOT exist for student ETU002 and subject 301.
  Assigned Subject ID 301 to Student Roll No: ETU002
Subject assigned to student successfully!
```

===== Student Management System (MongoDB) =====

```
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 4
Enter Student Roll No (e.g., STU001): STU003
Enter Student Name (e.g., Alice Smith): Alex Raj
Enter Semester (e.g., 1,2,3): 1
Enter Year of Admission (e.g., 2023): 2023
Enter Expected Passing Year (e.g., 2027): 2027
Enter Stream ID (e.g., 1 (Must match the stream ID you added earlier): 1
  Stream exists for ID: 1
  Student added: Alex Raj (Roll No: STU003)
Enter Contact Reg No (for this student) (e.g., CON001): CON005
Enter Phone (e.g., 1234567890): 4589145891
Enter Email (e.g., alice.s@example.com): alex@example.com
  Contact added for Roll No: STU003
Student and contact added successfully!
```

===== Student Management System (MongoDB) =====

```
1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit
Enter your choice: 6
```

=== All Students ===

Retrieved 5 students.

Roll No: STU001

Name: Alice Smith
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1
Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: STU001
Phone: 1234567890 | Email: alice@example.com
Retrieved 1 subjects for student STU001
Subjects: Subject found for ID: 201
Data Structures

Roll No: ST002
Name: Punit Jaiswal
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1
Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: ST002
Phone: 9813448134 | Email: punit@example.com
Retrieved 1 subjects for student ST002
Subjects: Subject found for ID: 202
RDBMS

Roll No: ETU001
Name: Harpreet Sukkhi
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 2
Stream: Electrical Engineering
Department found for ID: 102
Department: Electrical & Electronics Engineering
Contact found for Roll No: ETU001
Phone: 4564566789 | Email: harpreet@example.com
Retrieved 1 subjects for student ETU001
Subjects: Subject found for ID: 301
Fundamental Electrical & Electronics

Roll No: ETU002
Name: Gurmeet Singh
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 2
Stream: Electrical Engineering
Department found for ID: 102
Department: Electrical & Electronics Engineering
Contact found for Roll No: ETU002
Phone: 2345123451 | Email: gurmeet@example.com
Retrieved 1 subjects for student ETU002
Subjects: Subject found for ID: 301

Fundamental Electrical & Electronics

Roll No: STU003
Name: Alex Raj
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1
Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: STU003
Phone: 4589145891 | Email: alex@example.com
Retrieved 0 subjects for student STU003

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 9

Enter Student Roll No (e.g., STU001): STU003

Enter Subject ID to assign (e.g., 201): 201

Subject exists for ID: 201

Association does NOT exist for student STU003 and subject 201.

Assigned Subject ID 201 to Student Roll No: STU003

Subject assigned to student successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 6

=== All Students ===

Retrieved 5 students.

Roll No: STU001
Name: Alice Smith
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1

Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: STU001
Phone: 1234567890 | Email: alice@example.com
Retrieved 1 subjects for student STU001
Subjects: Subject found for ID: 201
Data Structures

Roll No: ST002
Name: Punit Jaiswal
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1
Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: ST002
Phone: 9813448134 | Email: punit@example.com
Retrieved 1 subjects for student ST002
Subjects: Subject found for ID: 202
RDBMS

Roll No: ETU001
Name: Harpreet Sukkhi
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 2
Stream: Electrical Engineering
Department found for ID: 102
Department: Electrical & Electronics Engineering
Contact found for Roll No: ETU001
Phone: 4564566789 | Email: harpreet@example.com
Retrieved 1 subjects for student ETU001
Subjects: Subject found for ID: 301
Fundamental Electrical & Electronics

Roll No: ETU002
Name: Gurmeet Singh
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 2
Stream: Electrical Engineering
Department found for ID: 102
Department: Electrical & Electronics Engineering
Contact found for Roll No: ETU002
Phone: 2345123451 | Email: gurmeet@example.com
Retrieved 1 subjects for student ETU002
Subjects: Subject found for ID: 301
Fundamental Electrical & Electronics

Roll No: STU003
Name: Alex Raj
Semester: 1

Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1
Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: STU003
Phone: 4589145891 | Email: alex@example.com
Retrieved 1 subjects for student STU003
Subjects: Subject found for ID: 201
Data Structures

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 7

Enter Roll No to Update Contact (e.g., STU001): ST002

Enter New Reg No (e.g., CON001-UPDATED): CON006

Enter New Phone (e.g., 0987654321): 8576185761

Enter New Email(e.g., alice.smith.new@example.com): punit@example.com

Contact for Roll No ST002 updated.

Contact updated successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 8

Enter Roll No to Delete (e.g., STU001): ST002

Attempting to delete student and associated records for Roll No: ST002

Contact for Roll No ST002 deleted.

Associated contact for ST002 deleted.

Deleted 1 subjects for student ST002.

Associated subject assignments for ST002 deleted.

Student with Roll No ST002 deleted successfully!

Student and associated records deleted successfully!

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject

4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 6

=== All Students ===
Retrieved 4 students.

Roll No: STU001
Name: Alice Smith
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 1
Stream: Software Engineering
Department found for ID: 101
Department: Computer Science
Contact found for Roll No: STU001
Phone: 1234567890 | Email: alice@example.com
Retrieved 1 subjects for student STU001
Subjects: Subject found for ID: 201
Data Structures

Roll No: ETU001
Name: Harpreet Sukkhi
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 2
Stream: Electrical Engineering
Department found for ID: 102
Department: Electrical & Electronics Engineering
Contact found for Roll No: ETU001
Phone: 4564566789 | Email: harpreet@example.com
Retrieved 1 subjects for student ETU001
Subjects: Subject found for ID: 301
Fundamental Electrical & Electronics

Roll No: ETU002
Name: Gurmeet Singh
Semester: 1
Admission Year: 2023
Passing Year: 2027
Stream found for ID: 2
Stream: Electrical Engineering
Department found for ID: 102
Department: Electrical & Electronics Engineering
Contact found for Roll No: ETU002
Phone: 2345123451 | Email: gurmeet@example.com
Retrieved 1 subjects for student ETU002
Subjects: Subject found for ID: 301
Fundamental Electrical & Electronics

```
Roll No: STU003
Name: Alex Raj
Semester: 1
Admission Year: 2023
Passing Year: 2027
  Stream found for ID: 1
Stream: Software Engineering
  Department found for ID: 101
Department: Computer Science
Contact found for Roll No: STU003
Phone: 4589145891 | Email: alex@example.com
  Retrieved 1 subjects for student STU003
Subjects:  Subject found for ID: 201
Data Structures
-----
```

===== Student Management System (MongoDB) =====

1. Add Department
2. Add Stream
3. Add Subject
4. Add Student with Contact
5. View Student Details by Roll No
6. View All Students
7. Update Student Contact
8. Delete Student by Roll No (Cascading)
9. Assign Subject to Student
10. Exit

Enter your choice: 10

Exiting application. Goodbye!

Every Change or insertion and Updation I made by interacting with the java is documented inside the mongo shell (so I will only paste the snippets of the mongo shell) please see the snippets for the reference purposes only:

MongoDB Shell Snippets (Verification): mongosh can be invoked from C:\Windows\System32> by typing mongosh.

```
C:\Windows\System32>mongosh
Current Mongosh Log ID: 684a9f3ccd557445ca50eb66
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:      8.0.10
Using Mongosh:      2.5.2
```

For mongosh info see: <https://www.mongodb.com/docs/mongodb-shell/>

```
-----
The server generated these startup warnings when booting
2025-06-12T02:20:20.405+05:30: Access control is not enabled for the database. Read and write access to data and configuration
-----
test> show dbs
```

Under **Show dbs** command will show the databases present inside the database

mongosh mongod://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
latest_db> use Students_db
switched to db Students_db
Students_db> db.show collections
Uncaught:
SyntaxError: Missing semicolon. (1:7)
```

```
> 1 | db.show collections
    | ^
    |
    2 |
```

```
Students_db> show collections
departments
Students_db> db.departments.find()
[
  {
    _id: ObjectId('604b0978aad62ee710391f77'),
    deptId: 101,
    deptName: 'Computer Science'
  }
]
```

```
Students_db> show collections
departments
Students_db> show collections
departments
Students_db> show collections
contacts
departments
streams
student_subjects
students
subjects
Students_db> db.contacts.find()
```

```
[
  {
    _id: ObjectId('604bc35f8ac720b301fe9b4b'),
    rollNo: 'STU001',
    regId: 'C0001',
    phone: '1234567890',
    email: 'alice@example.com'
  }
]
```

```
Students_db> db.departments.find()
[
  {
    _id: ObjectId('604b0978aad62ee710391f77'),
    deptId: 101,
    deptName: 'Computer Science'
  }
]
```

mongosh mongod://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
Students_db> db.departments.find()
[
  {
    _id: ObjectId('604b0978aad62ee710391f77'),
    deptId: 101,
    deptName: 'Computer Science'
  }
]
```

```
Students_db> db.streams.find()
[
  {
    _id: ObjectId('604bc2ac8ac720b301fe9b4b'),
    streamId: 1,
    streamName: 'Software Engineering',
    deptId: 101
  }
]
```

```
Students_db> db.student_subjects
Students_db.student_subjects
Students_db> db.students.find()
[
  {
    _id: ObjectId('604bc3378ac720b301fe9b4a'),
    rollNo: 'STU001',
    name: 'Alice Smith',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  }
]
```

```
Students_db> db.subjects.find()
[
  {
    _id: ObjectId('604bc2e28ac720b301fe9b4b'),
    subjectId: 201,
    subjectName: 'Data Structures'
  }
]
```

```
Students_db> db.streams.find()
[
  {
    _id: ObjectId('604bc2ac8ac720b301fe9b4b'),
    streamId: 1,
    streamName: 'Software Engineering',
    deptId: 101
  },
  {
    _id: ObjectId('604b09a30f4d46ec152191d'),

```

mongosh mongoDb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
{
  _id: ObjectId('684bd06a30fd646ec152191d'),
  streamId: 2,
  streamName: 'Electrical Engineering',
  deptId: 102
}
Students_db> db.subjects.find()
[
  {
    _id: ObjectId('684bc7e28ac720b301fe9b49'),
    subjectId: 201,
    subjectName: 'Data Structures'
  },
  {
    _id: ObjectId('684bd10130fd646ec152191e'),
    subjectId: 202,
    subjectName: 'RDBMS'
  },
  {
    _id: ObjectId('684bd2c830fd646ec1521922'),
    subjectId: 301,
    subjectName: 'Fundamental Electrical & Electronics'
  }
]
Students_db> db.students.find()
[
  {
    _id: ObjectId('684bc3378ac720b301fe9b4a'),
    rollNo: 'STU001',
    name: 'Alice Smith',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('684bd18230fd646ec152191f'),
    rollNo: 'STU002',
    name: 'Punit Jaiswal',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('684bd34730fd646ec1521923'),
    rollNo: 'TU0001',
    name: 'Harpreet Sukkhi',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('684bd42430fd646ec1521926'),
    rollNo: 'TU0002',
    name: 'Gurmeet Singh',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('684bd4cc30fd646ec1521929'),
    rollNo: 'STU003',
    name: 'Alex Raj',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  }
]
Students_db> db.students.find()
[
  {
    _id: ObjectId('684bc3378ac720b301fe9b4a'),
    rollNo: 'STU001',
    name: 'Alice Smith',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('684bd18230fd646ec152191f'),
    rollNo: 'STU002',
    name: 'Punit Jaiswal',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('684bd34730fd646ec1521923'),
    rollNo: 'TU0001',
    name: 'Harpreet Sukkhi',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('684bd42430fd646ec1521926'),
    rollNo: 'TU0002',
    name: 'Gurmeet Singh',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('684bd4cc30fd646ec1521929'),
    rollNo: 'STU003',
    name: 'Alex Raj',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  }
]
```

mongosh mongoDb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
semester: 1,
yearOfAdmission: 2023,
expectedPassingYear: 2027,
streamId: 2
},
{
  _id: ObjectId('684bd42430fd646ec1521926'),
  rollNo: 'TU0002',
  name: 'Gurmeet Singh',
  semester: 1,
  yearOfAdmission: 2023,
  expectedPassingYear: 2027,
  streamId: 2
},
{
  _id: ObjectId('684bd4cc30fd646ec1521929'),
  rollNo: 'STU003',
  name: 'Alex Raj',
  semester: 1,
  yearOfAdmission: 2023,
  expectedPassingYear: 2027,
  streamId: 1
}
]
Students_db> db.students.find()
[
  {
    _id: ObjectId('684bc3378ac720b301fe9b4a'),
    rollNo: 'STU001',
    name: 'Alice Smith',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('684bd18230fd646ec152191f'),
    rollNo: 'STU002',
    name: 'Punit Jaiswal',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('684bd34730fd646ec1521923'),
    rollNo: 'TU0001',
    name: 'Harpreet Sukkhi',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('684bd42430fd646ec1521926'),
    rollNo: 'TU0002',
    name: 'Gurmeet Singh',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('684bd4cc30fd646ec1521929'),
    rollNo: 'STU003',
    name: 'Alex Raj',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  }
]
```


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
semester: 1,
yearOfAdmission: 2023,
expectedPassingYear: 2027,
streamId: 1
}
]
Students_db> db.contacts.find()
[
  {
    _id: ObjectId('604bc35f8ac720b301fe9b4b'),
    rollNo: 'STU001',
    regNo: 'CO0001',
    phone: '1234567890',
    email: 'alice@example.com'
  },
  {
    _id: ObjectId('604bd1ea30fd646ec1521920'),
    rollNo: 'STU002',
    regNo: 'CO0006',
    phone: '8978185701',
    email: 'pun1@example.com'
  },
  {
    _id: ObjectId('604bd3a130fd646ec1521924'),
    rollNo: 'ETU0001',
    regNo: 'CO0003',
    phone: '456456789',
    email: 'harpreet@example.com'
  },
  {
    _id: ObjectId('604bd45730fd646ec1521927'),
    rollNo: 'ETU0002',
    regNo: 'CO0004',
    phone: '2345123451',
    email: 'gurmeet@example.com'
  },
  {
    _id: ObjectId('604bd50830fd646ec152192a'),
    rollNo: 'STU003',
    regNo: 'CO0005',
    phone: '4589145891',
    email: 'alex@example.com'
  }
]
Students_db> db.students.find()
[
```

```
  {
    _id: ObjectId('604bc3378ac720b301fe9b4a'),
    rollNo: 'STU001',
```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
rollNo: 'STU003',
regNo: 'CO0005',
phone: '4589145891',
email: 'alex@example.com'
}
]
Students_db> db.students.find()
[
  {
    _id: ObjectId('604bc3378ac720b301fe9b4a'),
    rollNo: 'STU001',
    name: 'Alice Smith',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  },
  {
    _id: ObjectId('604bd34730fd646ec1521923'),
    rollNo: 'ETU0001',
    name: 'Harpreet Sukhhi',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('604bd42430fd646ec1521926'),
    rollNo: 'ETU0002',
    name: 'Gurmeet Singh',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 2
  },
  {
    _id: ObjectId('604bd4ec30fd646ec1521929'),
    rollNo: 'STU003',
    name: 'Alex Raj',
    semester: 1,
    yearOfAdmission: 2023,
    expectedPassingYear: 2027,
    streamId: 1
  }
]
Students_db> db.contacts.find()
[
  {
    _id: ObjectId('604bc35f8ac720b301fe9b4b'),
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
{
  _id: ObjectId('684bd4cc30fd646ec1521929'),
  rollNo: 'ET0001',
  name: 'Alice Raj',
  semester: 1,
  yearOfAdmission: 2023,
  expectedPassingYear: 2027,
  streamId: 1
}
Students_db> db.contacts.find()
[
  {
    _id: ObjectId('684bd35f8ac720b301fe9b4b'),
    rollNo: 'ET0001',
    regNo: 'C00001',
    phone: '1234567890',
    email: 'alice@example.com'
  },
  {
    _id: ObjectId('684bd3a130fd646ec1521924'),
    rollNo: 'ET0001',
    regNo: 'C00003',
    phone: '4564566789',
    email: 'harpreet@example.com'
  },
  {
    _id: ObjectId('684bd45730fd646ec1521927'),
    rollNo: 'ET0002',
    regNo: 'C00004',
    phone: '2343234561',
    email: 'gurmeet@example.com'
  },
  {
    _id: ObjectId('684bd50030fd646ec152192a'),
    rollNo: 'ET0001',
    regNo: 'C00005',
    phone: '4569145691',
    email: 'alex@example.com'
  }
]
Students_db> db.subjects.find().count()
3
Students_db> db.contacts.find().count()
4
Students_db> db.students.find().count()
4
Students_db> db.streams.find().count()
2
```

```
Administrator: Command Pr...
Students_db> Exit
ReferenceError: Exit is not defined
Students_db> terminate
ReferenceError: terminate is not defined
Students_db> purge()
ReferenceError: purge is not defined
Students_db> quit

C:\Windows\System32>
```

Usage of the quit command to exit the mongoshell terminal.

The output from the Java console and mongosh terminal can be cross-referenced to observe the real-time data changes within the database. Use the quit command to exit mongosh.

The file structure will look like

Future Scope:

This V1.0 project lays a robust foundation for a comprehensive Student Management System. Future enhancements and potential features include:

Cloud Hosting & CI/CD:

- Transitioning the MongoDB database to a cloud cluster (e.g., MongoDB Atlas) for enhanced scalability, reliability, and accessibility.
- Implementing a Continuous Integration/Continuous Delivery (CI/CD) pipeline to automate build, test, and deployment processes, ensuring faster and more reliable software delivery.

Extended Data Models:

- Incorporating additional data fields and entities as required by specific educational institution needs (e.g., attendance records, grades, course schedules, faculty information).

Scalability Enhancements:

- Exploring and implementing horizontal scaling strategies to manage increasing data volumes and user loads efficiently.
- Optimizing data handling and overall application efficiency by selecting appropriate MongoDB Atlas cluster tiers to match performance and cost requirements.
- Facilitating the insertion of diverse data types, including multimedia (pictures), geographical information, and complex financial records.

Graphical User Interface (GUI) Integration:

- Developing an intuitive and interactive front-end using modern web technologies such as React.js, Tailwind CSS, HTML, and JavaScript, coupled with a Java Spring Boot Framework for the backend API.

Security & Access Control:

- Implementing robust login and logout security features, potentially leveraging external authentication services like Google Authenticator.
- Establishing granular role-based access control (RBAC) directly within MongoDB to define and enforce permissions, restricting lower-level users from unauthorized viewing, modification, deletion, or sharing of sensitive, higher-level data.

Conclusion

The V1.0 of this Student Management System backend project, utilizing Java and MongoDB, successfully demonstrates the efficient handling of fundamental student-related data operations. This initial build provides a solid, functional core with essential features implemented. The architecture supports future expansion, allowing for the progressive integration of more complex functionalities, scalability improvements, and a richer user experience in subsequent versions.