

MongoDB Project - Screenshots

Name: Mekapothula Durga Venkateswara Reddy

Reg.no: 22BCB7296

Gamil: venkateswara.22bcb7296@vitapstudent.ac.in

Project Title: Full-Stack Student & Course Management System

Description:

The current project is a full-stack web application that allows educational institutions to manage students and course data efficiently. It's built on HTML, CSS, JavaScript (in the front-end), Node.js (in the back end) and MongoDB (for the database). In addition to all CRUD (Create, Read, Update, Delete) operations on students and courses performed by the admin, for it is a full-stack web application, it also has easy navigation, a responsive user interface, as well as an enrolment tracking and status tracking feature, dynamic search for students and courses, among many other functions. It increases efficiency for academic record keeping allowing one central place for school, college or training center to glean information in real time, all the time on its students and courses.

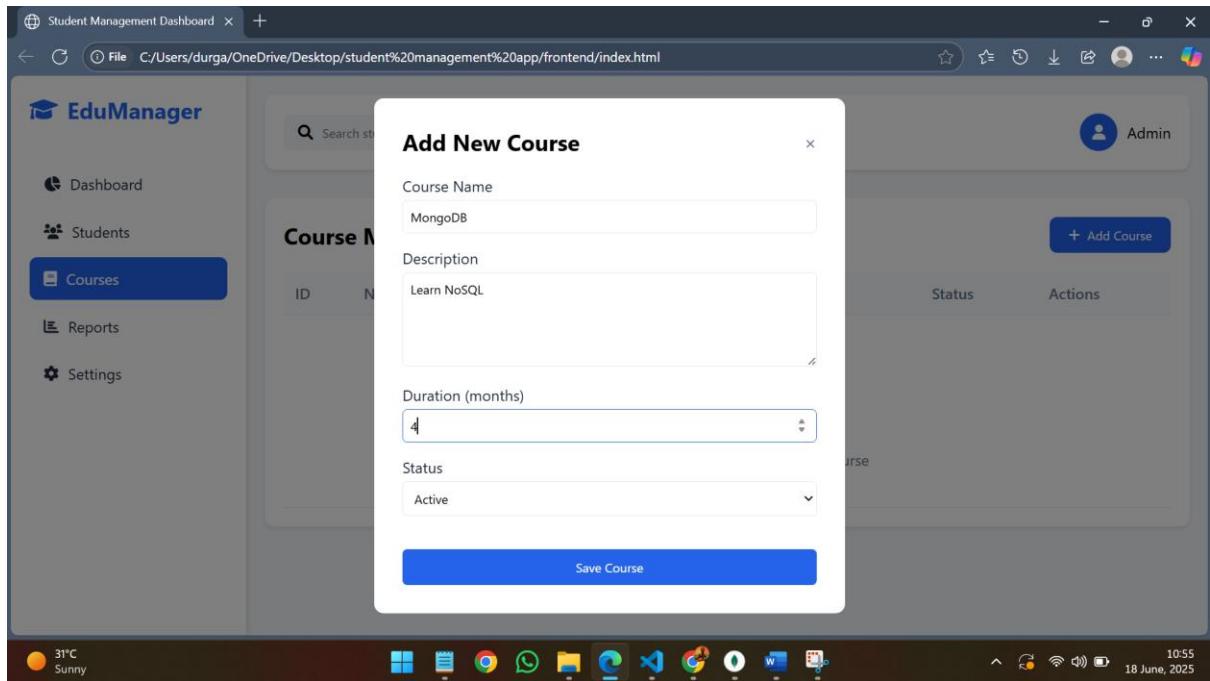
1. Initial view of web page.

The screenshot shows the EduManager Student Management Dashboard. On the left, there's a sidebar with a logo and navigation links: Dashboard, Students, Courses (which is highlighted in blue), Reports, and Settings. The main area has a search bar at the top. Below it are four cards: 'Total Students' (0), 'Active Courses' (0), 'Graduates' (0), and 'Success Rate' (0%). A section titled 'Recent Students' shows a table with columns: ID, Name, Course, Enrollment Date, Status, and Actions. A message below the table says 'No Students Found' and suggests adding a student. The bottom status bar shows the weather as 31°C Sunny, the date as 18 June, 2025, and the time as 10:51.

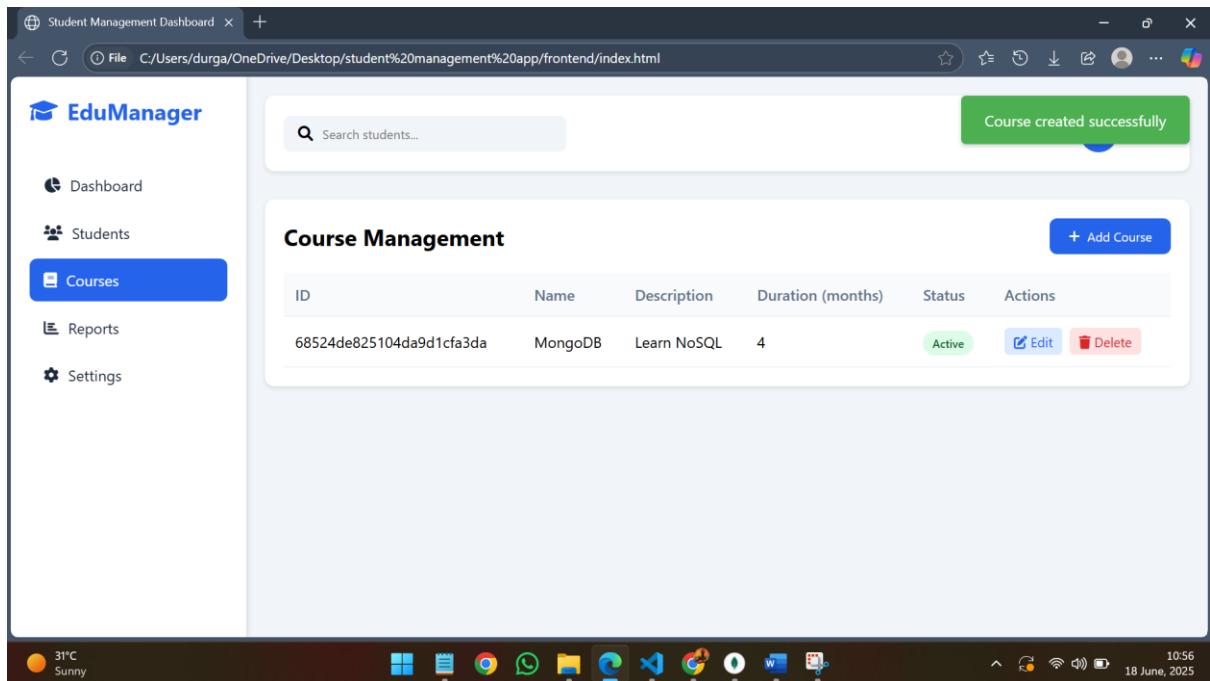
2. Web page notifying to add the course before adding the students.

The screenshot shows the EduManager Student Management Dashboard with a modal window open. The modal is titled 'Add New Course' and contains fields for 'Course Name', 'Description', 'Duration (months)', and 'Status' (set to 'Active'). A yellow banner at the top right of the modal says 'Please add at least one course before adding students'. In the background, the sidebar shows 'Courses' is selected. The bottom status bar shows the weather as 31°C Sunny, the date as 18 June, 2025, and the time as 10:53.

3.Adding the course details.



4.Web page notifying that course is added successfully.



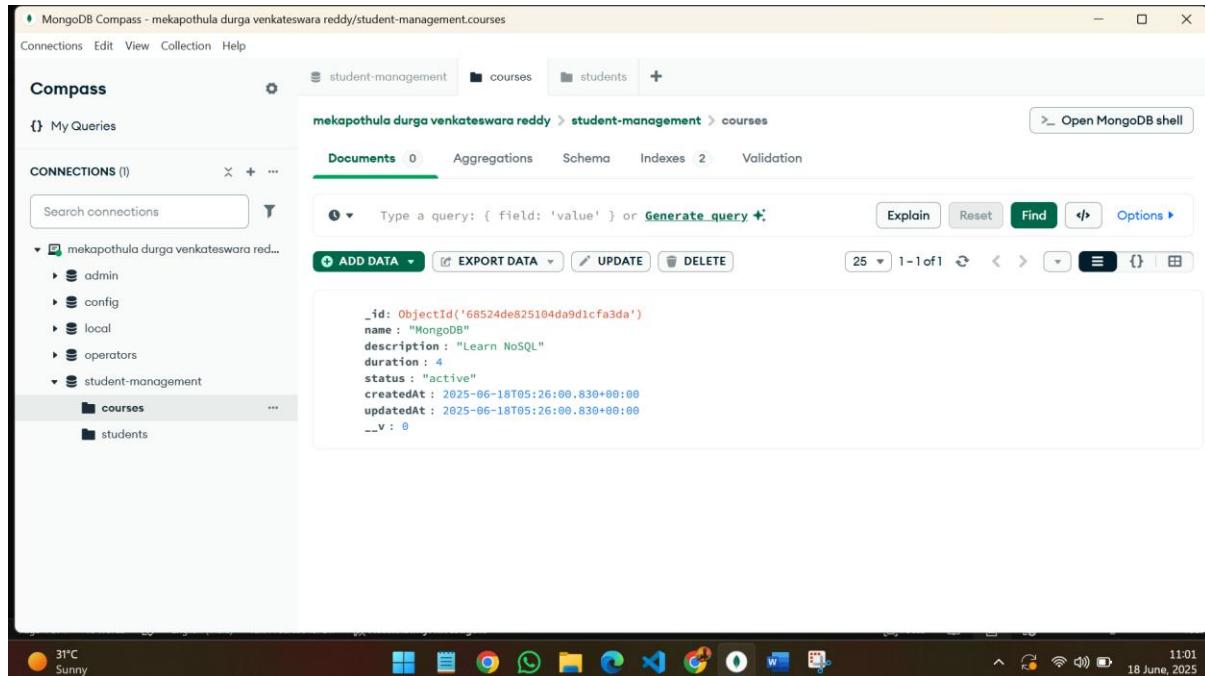
5. Initial view of compass.

The screenshot shows the MongoDB Compass interface. At the top, there's a navigation bar with 'Connections', 'Edit', 'View', and 'Help'. Below it, the title bar says 'MongoDB Compass - mekapothula durga venkateswara reddy/student-management'. The main area is titled 'Compass' and shows 'My Queries'. On the left, under 'CONNECTIONS (1)', there's a list of databases: 'mekapothula durga venkateswara reddy' (with sub-options 'admin', 'config', 'local', 'operators'), and 'student-management' (with sub-options 'courses', 'students'). In the center, there are two sections: 'courses' and 'students'. Both sections show storage size (8.19 kB and 4.10 kB respectively), document count (0), average document size (0 B), index count (2), and total index size (24.58 kB and 8.19 kB). At the bottom right, there's a status bar showing '31°C Sunny', the date '18 June, 2025', and the time '10:58'.

6. Empty course collection before adding the course.

This screenshot shows the 'courses' collection within the 'student-management' database in MongoDB Compass. The title bar indicates 'MongoDB Compass - mekapothula durga venkateswara reddy/student-management.courses'. The main area has tabs for 'Documents' (selected), 'Aggregations', 'Schema', 'Indexes', and 'Validation'. Under 'Documents', it says '0'. There are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. A message at the bottom states 'This collection has no data' and suggests importing from a JSON or CSV file. A 'Import Data' button is visible. The left sidebar shows the 'student-management' connection with 'courses' selected. The bottom right shows the same system status as the previous screenshot.

7.After refreshing we can see the course added in the web page in compass.

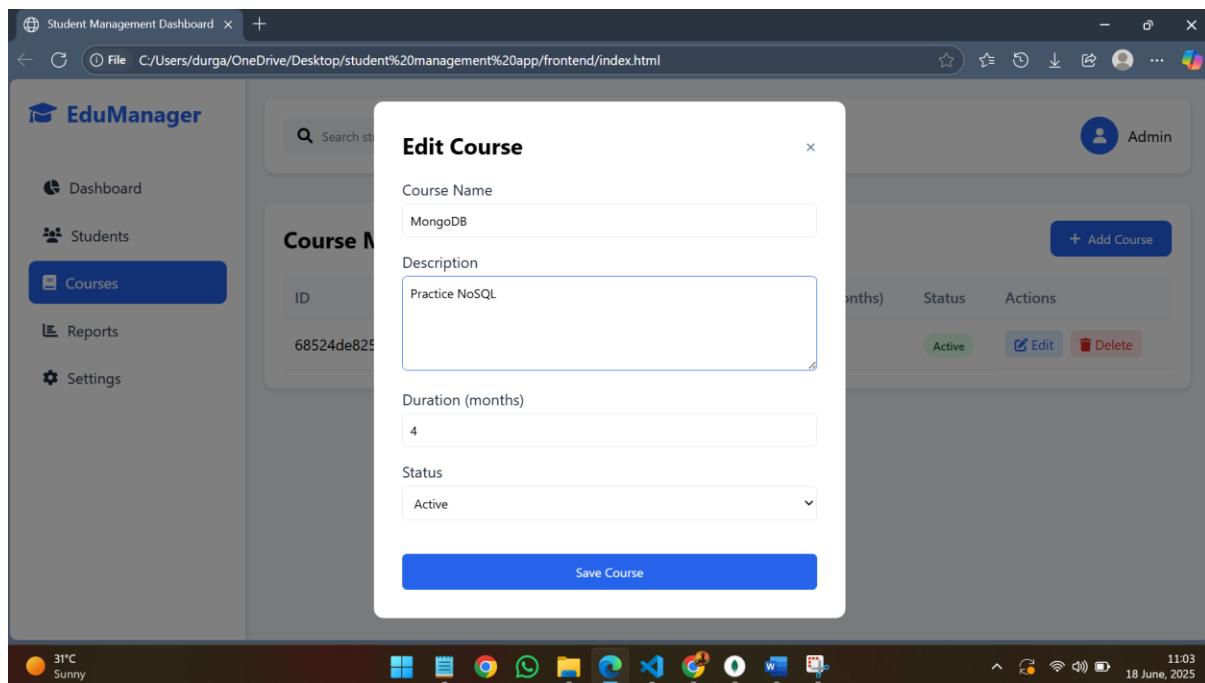


The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'mekapothula durga venkateswara reddy' and its databases: 'student-management', 'courses', and 'students'. The main panel displays the 'Courses' collection with one document listed:

```
_id: ObjectId('68524de825104da9d1cfa3da')
name : "MongoDB"
description : "Learn NoSQL"
duration : 4
status : "active"
createdAt : 2025-06-18T05:26:00.830+00:00
updatedAt : 2025-06-18T05:26:00.830+00:00
__v : 0
```

The status bar at the bottom shows it's 11:01 on June 18, 2025.

8.Editing the course details in web page.



The screenshot shows the EduManager web application. The left sidebar has 'Courses' selected. The main area shows a list of courses with one highlighted. A modal window titled 'Edit Course' is open, containing the following fields:

- Course Name: MongoDB
- Description: Practice NoSQL
- Duration (months): 4
- Status: Active

A blue 'Save Course' button is at the bottom of the modal. The status bar at the bottom shows it's 11:03 on June 18, 2025.

9.Successful update of course details notification in web page.

The screenshot shows the 'Course Management' section of the EduManager application. A green success message 'Course updated successfully' is displayed at the top right. The table lists one course entry:

ID	Name	Description	Duration (months)	Status	Actions
68524de825104da9d1cfa3da	MongoDB	Practice NoSQL	4	Active	Edit Delete

10.Trying to delete the course in web page.

The screenshot shows the 'Course Management' section with a modal dialog titled 'Confirm Deletion'. The dialog contains the message: 'Are you sure you want to delete this item? This action cannot be undone.' with 'Cancel' and 'Delete' buttons. The background table shows the same course entry as before.

ID	Name	Description	Duration (months)	Status	Actions
68524de825104da9d1cfa3da	MongoDB	Practice NoSQL	4	Active	Edit Delete

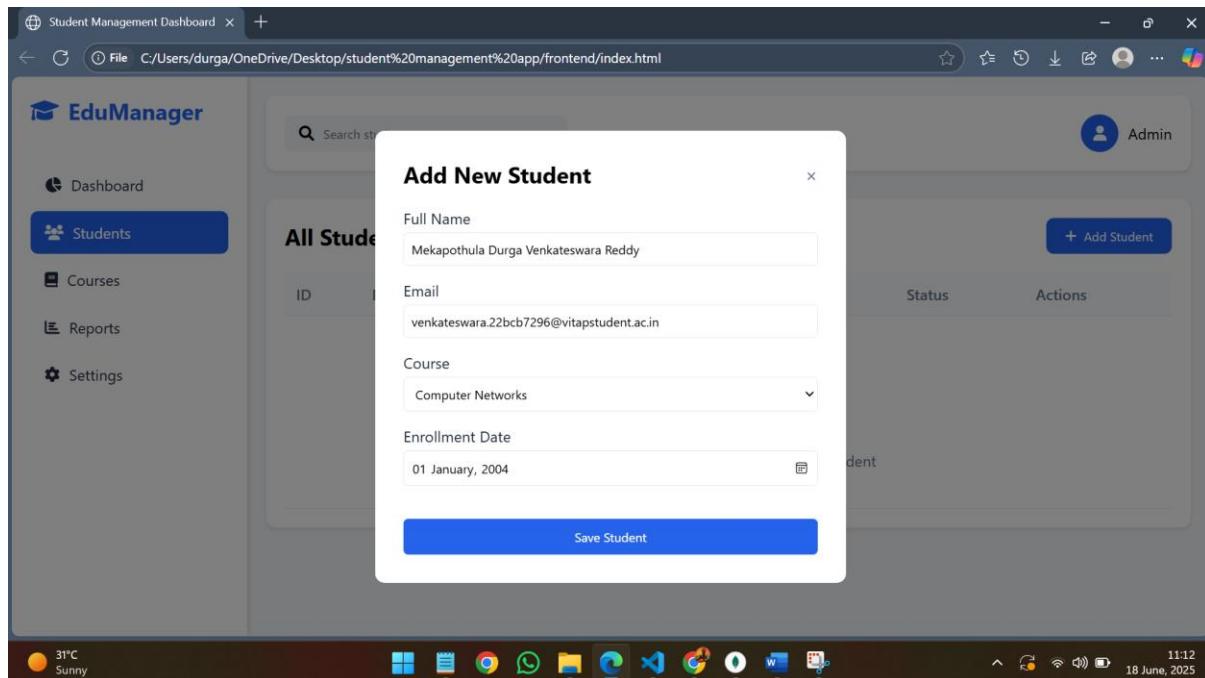
11.Notification of “course deleted successfully” in web page.

The screenshot shows the EduManager Student Management Dashboard. On the left, there's a sidebar with links: Dashboard, Students, Courses (which is highlighted in blue), Reports, and Settings. The main area is titled "Course Management". It has a search bar at the top with placeholder text "Search students...". A green button on the right says "Course deleted successfully". Below the search bar, there's a table header with columns: ID, Name, Description, Duration (months), Status, and Actions. Underneath the table, a message says "No Courses Found" and "Click "Add Course" to add your first course". The bottom of the screen shows a Windows taskbar with various icons and the date/time: 18 June, 2025, 11:07.

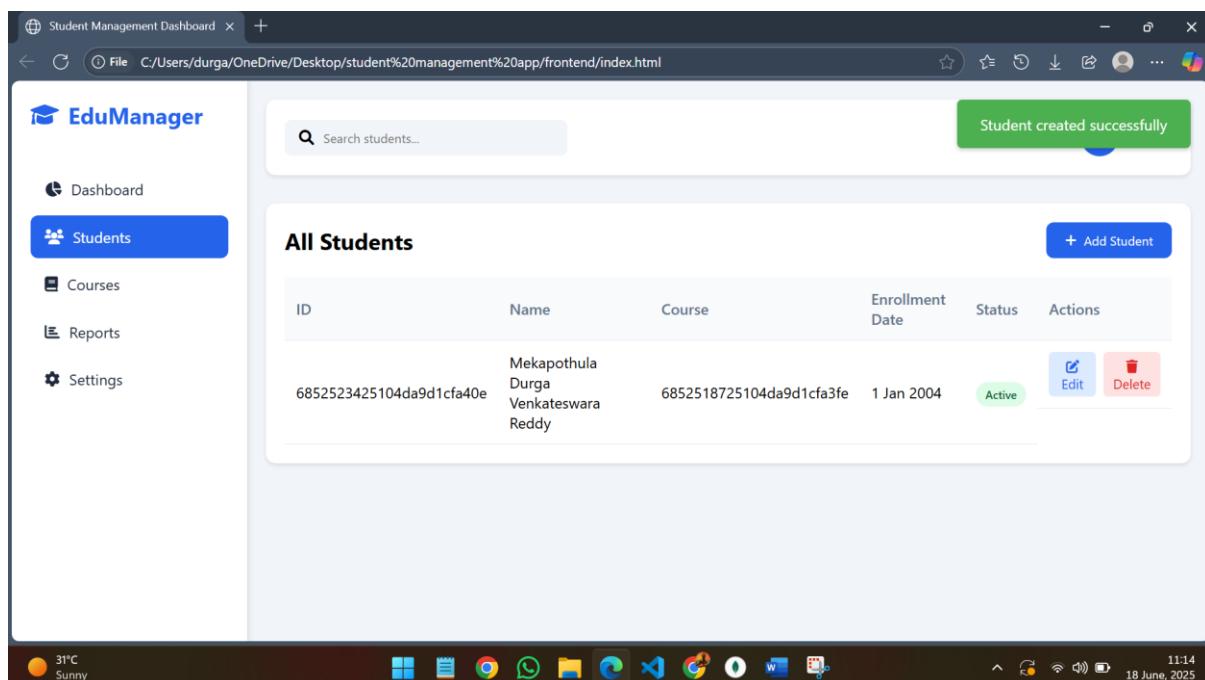
12.Now if compass is refreshed course is also deleted from the MongoDB.

The screenshot shows the MongoDB Compass interface. The title bar says "MongoDB Compass - mekapothula durga venkateswara reddy/student-management.courses". The left sidebar shows connections: "mekapothula durga venkateswara reddy" (selected) and "student-management" (selected). Under "student-management", there are collections: "courses" (selected) and "students". The main panel shows the "courses" collection. It has tabs for "Documents" (0), "Aggregations", "Schema", "Indexes" (2), and "Validation". Below these tabs, there's a search bar with placeholder text "Type a query: { field: 'value' } or Generate query +". There are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". A status bar at the bottom says "25 0 - 0 of 0". In the center, there's a small icon of a document with a dashed border. Below the icon, a message says "This collection has no data". At the bottom, there's a "Import Data" button. The bottom of the screen shows a Windows taskbar with various icons and the date/time: 18 June, 2025, 11:09.

13.If any course is present now students can be added. Filling the student details.



14." student created successfully" notification from the web page.



15. Now we can see the added student details in the compass.

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists a connection named 'mekapothula durga venkateswara reddy'. Under this connection, the 'student-management' database is selected, and its 'students' collection is shown. The main pane displays a single document with the following fields and values:

```
_id: ObjectId('6852523425104da9d1cfa40e')
name : "Mekapothula Durga Venkateswara Reddy"
email : "venkateswara.22bcb7296@vitapstudent.ac.in"
course : "6852518725104da9d1cfa3fe"
enrollmentDate : 2004-01-01T00:00:00+00:00
status : "active"
createdAt : 2025-06-18T05:44:20.303+00:00
updatedAt : 2025-06-18T05:44:20.303+00:00
__v : 0
```

16. Editing the student details in the web page.(course is edited in student record)

The screenshot shows a web browser window titled 'Student Management Dashboard'. The URL is 'C:/Users/durga/OneDrive/Desktop/student%20management%20app/frontend/index.html'. The dashboard has a sidebar with 'EduManager' branding and links for 'Dashboard', 'Students' (which is highlighted in blue), 'Courses', 'Reports', and 'Settings'. The main content area shows a table of 'All Students' with columns for ID, Enrollment Date, Status, and Actions. One row is selected, showing the ID '6852523425'. A modal dialog box titled 'Edit Student' is open over the table, containing the following form fields:

Full Name	Mekapothula Durga Venkateswara Reddy
Email	venkateswara.22bcb7296@vitapstudent.ac.in
Course	MongoDB
Enrollment Date	01 January, 2004

At the bottom of the modal is a blue 'Save Student' button.

17." student updated successfully" notification in the web page.

The screenshot shows a web browser window titled "Student Management Dashboard". The URL is "C:/Users/durga/OneDrive/Desktop/student%20management%20app/frontend/index.html". On the left, there's a sidebar with "EduManager" logo and navigation links: Dashboard, Students (highlighted in blue), Courses, Reports, and Settings. The main area has a search bar "Search students..." and a green button "Student updated successfully". Below it is a table titled "All Students" with columns: ID, Name, Course, Enrollment Date, Status, and Actions. A single student record is listed:

ID	Name	Course	Enrollment Date	Status	Actions
6852523425104da9d1cfa40e	Mekapothula Durga Venkateswara Reddy	685253fd25104da9d1cfa419	1 Jan 2004	Active	<button>Edit</button> <button>Delete</button>

The status bar at the bottom shows "31°C Sunny" and the date "18 June, 2025".

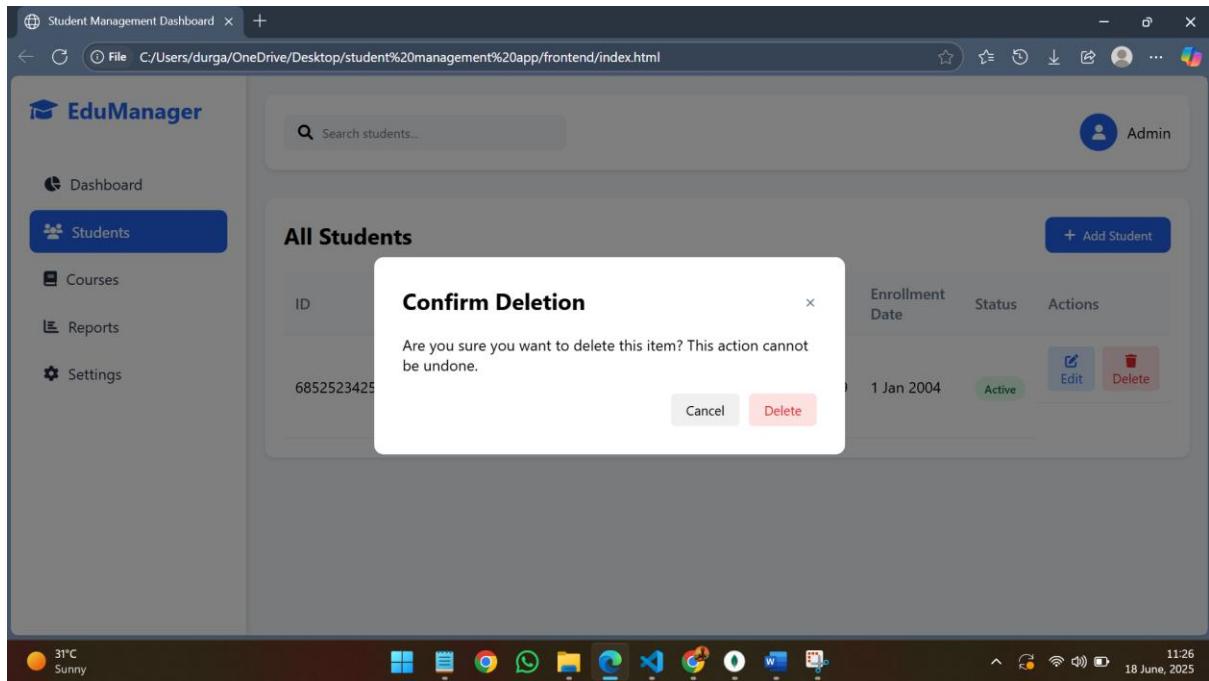
18.We can see the updated course id in the compass after updating the course in the student record as shown above.

The screenshot shows the MongoDB Compass interface. The connection is "mekapothula durga venkateswara reddy" and the database is "student-management". The collection is "students". The document details are as follows:

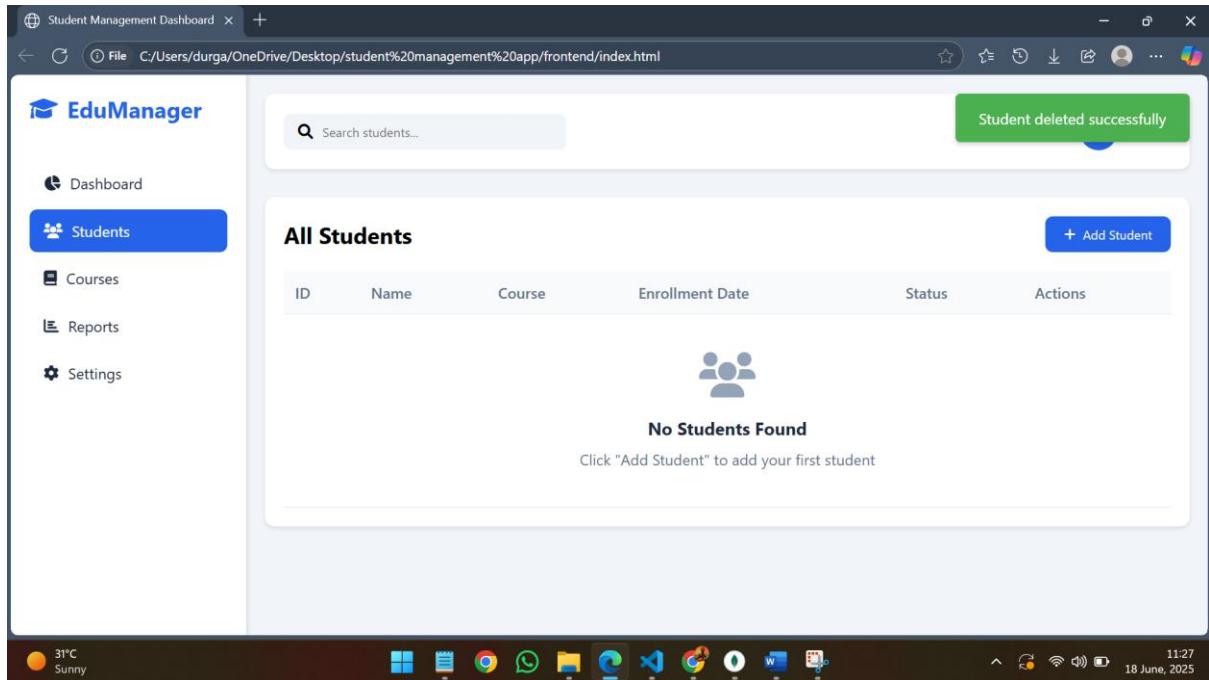
```
_id: ObjectId('6852523425104da9d1cfa40e')
name: "Mekapothula Durga Venkateswara Reddy"
email: "venkateswara.22bcb7296@vitapstudent.ac.in"
course: "685253fd25104da9d1cfa419"
enrollmentDate: 2004-01-01T00:00:00.000+00:00
status: "active"
createdAt: 2025-06-18T05:44:20.303+00:00
updatedAt: 2025-06-18T05:53:38.831+00:00
__v: 0
```

The status bar at the bottom shows "31°C Sunny" and the date "18 June, 2025".

19.Trying to delete the student record from the web page.



20." student deleted successfully" notification in the web page.



21.student collection is empty int the compass, after deleting the student record in the web page.

The screenshot shows the MongoDB Compass interface. The title bar reads "MongoDB Compass - mekapothula durga venkateswara reddy/student-management.students". The left sidebar is titled "Compass" and shows "CONNECTIONS (1)" with a list of databases: admin, config, local, operators, and student-management, with "students" selected. The main panel is titled "mekapothula durga venkateswara reddy > student-management > students". It has tabs for "Documents" (0), Aggregations, Schema, Indexes (2), and Validation. Below these are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". A status bar at the bottom indicates "0 - 0 of 0". In the center, there is a small icon of a document with a green checkmark. Below it, the text "This collection has no data" is displayed, followed by the note "It only takes a few seconds to import data from a JSON or CSV file." and a "Import Data" button. The system tray at the bottom shows the date as 18 June, 2025.

22.Searching a particular student with name.

The screenshot shows the "Student Management Dashboard" of the EduManager application. The title bar reads "Student Management Dashboard". The left sidebar has navigation links: "Dashboard", "Students" (which is highlighted in blue), "Courses", "Reports", and "Settings". The main content area is titled "All Students" and contains a table with the following data:

ID	Name	Course	Enrollment Date	Status	Actions
6852562725104da9d1cfa44c	Balaji	6852518725104da9d1cfa3fe	2 Feb 2004	Active	Edit Delete

A search bar at the top right contains the text "bal". The system tray at the bottom shows the date as 18 June, 2025.

23.In the dashboard we can see the total number of students and active courses.

The screenshot shows the EduManager Student Management Dashboard. On the left, there's a sidebar with a 'Dashboard' button (which is highlighted in blue) and other options: Students, Courses, Reports, and Settings. The main area has a search bar at the top labeled 'Search students...'. Below it are three cards: 'Total Students' (7), 'Active Courses' (2), and 'Graduates' (0). Further down is a card for 'Success Rate' (0%). At the bottom, there's a section titled 'Recent Students' with a table header (ID, Name, Course, Enrollment Date, Status, Actions) and a single row of data. A blue '+ Add Student' button is located in the top right of this section. The system status bar at the bottom indicates it's 11:38 on June 18, 2025, with a weather icon showing 33°C and sunny conditions.

24.Editing the student record in compass.

The screenshot shows the MongoDB Compass interface. The left sidebar lists connections: 'mekapothula durga venkateswara reddy' (selected), 'admin', 'config', 'local', 'operators', and 'student-management' (which contains 'courses' and 'students'). In the main pane, under the 'students' collection, a document is selected. The document details are as follows:

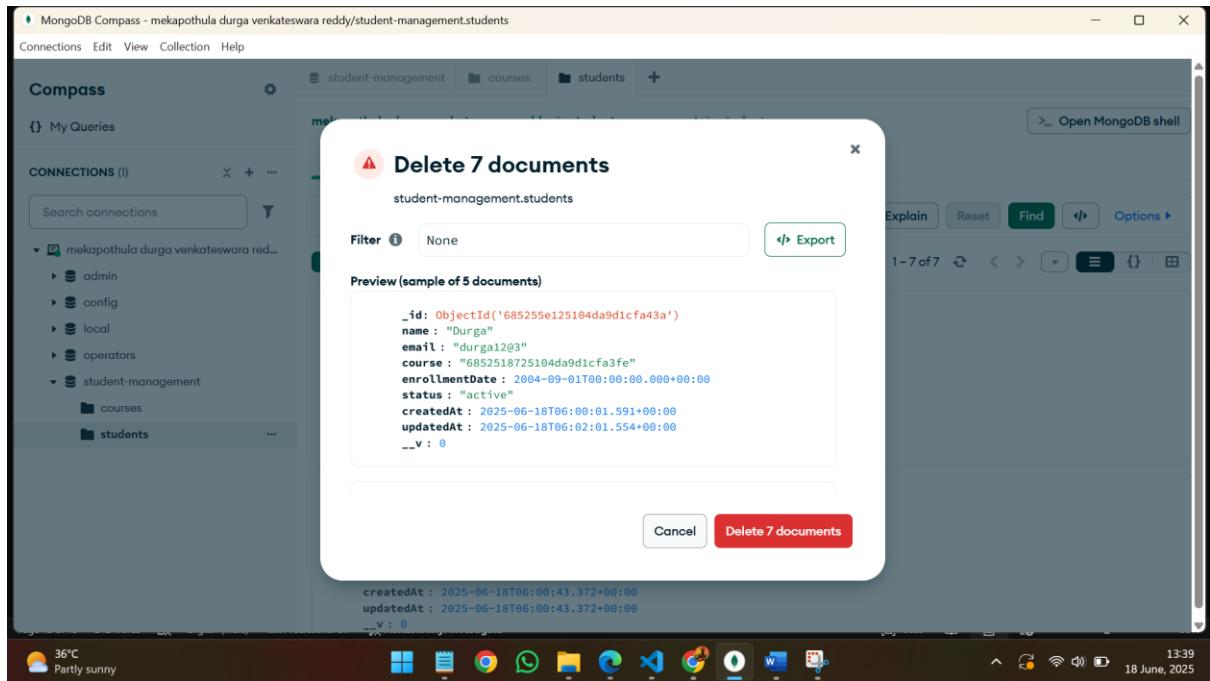
Field	Type
_id	ObjectId
name	String
email	String
course	String
enrollmentDate	Date
status	String
createdAt	Date
updatedAt	Date
__v	Int32

The document itself shows the following data:

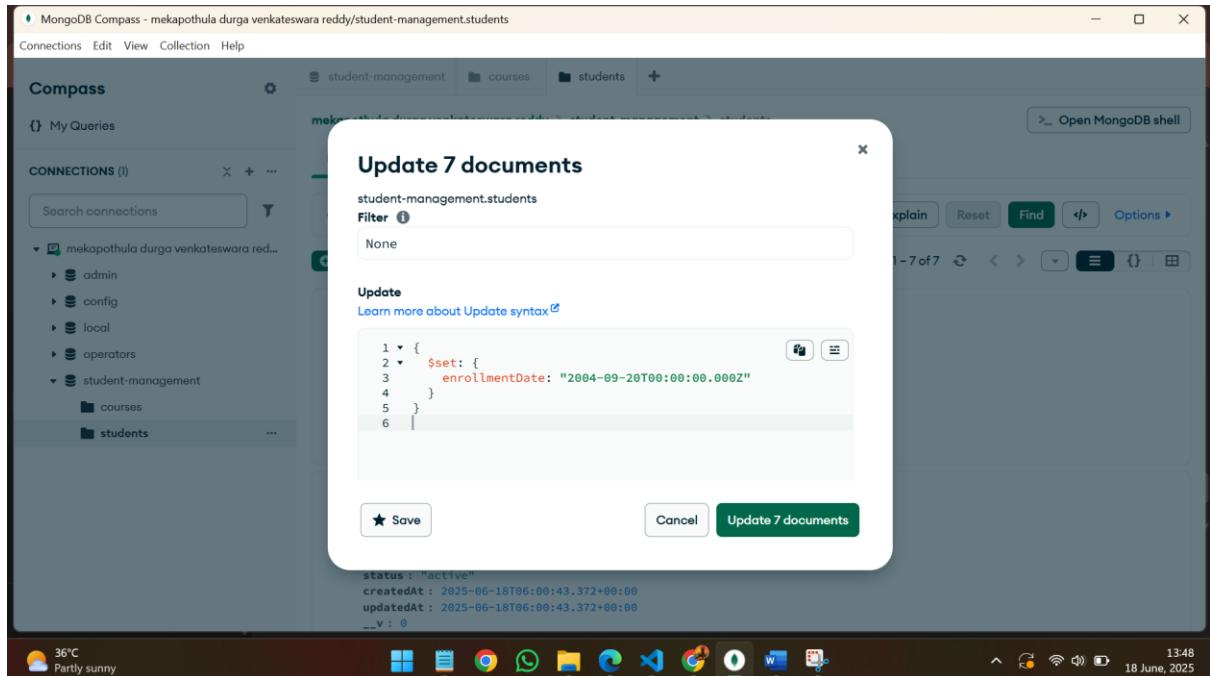
```
_id: ObjectId('685255e125104da9d1cfa43a')
name: "Durga"
email: "durga123@gmail.com"
course: "6852518725104da9d1cfa3fe"
enrollmentDate: 2004-09-01T00:00:00.000+00:00
status: "active"
createdAt: 2025-06-18T06:00:01.591+00:00
updatedAt: 2025-06-18T06:02:01.554+00:00
__v: 0
```

A yellow banner at the bottom indicates 'Document modified.' with 'CANCEL' and 'UPDATE' buttons. The system status bar at the bottom shows it's 13:36 on June 18, 2025, with a weather icon showing 33°C and hot weather.

25.Trying to delete the student records in compass.



26.Updating all documents in students collection in compass.



27.The update in the compass is reflected in the web page, changing all enrolment dates to 20 sept 2004.

The screenshot shows the EduManager Student Management Dashboard. On the left, there's a sidebar with 'Dashboard', 'Students' (which is highlighted in blue), 'Courses', 'Reports', and 'Settings'. The main area is titled 'All Students' and lists seven student entries. Each entry includes fields for ID, Name, Course, Enrollment Date, Status, and Actions (Edit and Delete buttons). The 'Enrollment Date' column for all students shows '20 Sept 2004'. The status for all students is 'Active'. The dashboard also features a search bar at the top and a weather widget on the bottom left indicating '36°C Partly sunny'.

28.Inserting new documents in students' collection in compass.

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'mekapothula durga venkateswara reddy/student-management.students'. The main area shows the 'student-management' database with 'courses' and 'students' collections. A modal window titled 'Insert Document' is open, prompting to insert into the 'student-management.students' collection. The document being inserted is a JSON object:

```
1 > {  
2   "_id": { "$oid": "685277abecdd9de38664f9f" },  
3   "name": "saketh",  
4   "email": "saketh837@example.com",  
5   "course": { "$oid": "6852518725104da9d1cfa3fe" },  
6   "enrollmentDate": { "$date": "2004-09-12T00:00:00Z" },  
7   "status": "active",  
8   "createdAt": { "$date": "2025-06-18T06:30:00.000Z" },  
9   "updatedAt": { "$date": "2025-06-18T06:30:00.000Z" },  
10  "__v": 0  
11 }  
12
```

The modal also shows the status, created at, and updated at fields. At the bottom right of the modal are 'Cancel' and 'Insert' buttons. The status bar at the bottom indicates '13:56 18 June, 2025'.

29. Newly inserted documents are now visible in the web page.

The screenshot shows the 'EduManager' student management system. On the left sidebar, under the 'Students' tab, there is a list of students. The main area displays a table titled 'All Students' with columns: ID, Name, Course, Enrollment Date, Status, and Actions. There are four student entries:

ID	Name	Course	Enrollment Date	Status	Actions
685277abaecd9de38664f9f	jakka venkata sai akhil	6852518725104da9d1cfa3fe	12 Sept 2004	Active	<button>Edit</button> <button>Delete</button>
685278f4aecdd9de38664fa3	Saketh	6852518725104da9d1cfa3fe	12 Sept 2004	Active	<button>Edit</button> <button>Delete</button>
6852571b25104da9d1cfa498	kashyap	6852518725104da9d1cfa3fe	20 Sept 2004	Active	<button>Edit</button> <button>Delete</button>
685256fd25104da9d1cfa48f	madhu	685253fd25104da9d1cfa419	20 Sept 2004	Active	<button>Edit</button> <button>Delete</button>

30. Finding the specific documents from the student collection.

Like email id containing letter 'd'.

The screenshot shows the MongoDB Compass interface. In the 'Documents' tab, a search query is defined: { "email": { "\$regex": "d", "\$options": "i" } }. Below the query, two student documents are listed:

```
_id: ObjectId('685255e125104da9d1cfa43a')
name: "Durga"
email: "durga12@3"
course: "6852518725104da9d1cfa3fe"
enrollmentDate: "2004-09-20T00:00:00.000Z"
status: "active"
createdAt: 2025-06-18T06:00:01.591+00:00
updatedAt: 2025-06-18T06:02:01.554+00:00
__v: 0

_id: ObjectId('685256fd25104da9d1cfa48f')
name: "madhu"
email: "madhu@45"
course: "685253fd25104da9d1cfa419"
enrollmentDate: "2004-09-20T00:00:00.000Z"
status: "active"
createdAt: 2025-06-18T06:04:45.241+00:00
updatedAt: 2025-06-18T06:04:45.241+00:00
__v: 0
```

31.node.js connection.

The screenshot shows the Visual Studio Code interface with the title bar "student management app". The Explorer sidebar on the left shows a project structure with "STUDENT MANAGEMENT APP" expanded, revealing "backend", "node_modules", ".env", "combined.log", "error.log", "package-lock.json", "package.json", and "server.js". The "server.js" file is selected in the Explorer. The Terminal tab at the top has "TERMINAL" selected. The terminal window displays the command "npm init -y" being run in the "backend" directory, followed by the generated package.json content. The status bar at the bottom right shows the date and time as "18 June, 2025".

```
C:\Users\durga\OneDrive\Desktop\student management app>cd backend
C:\Users\durga\OneDrive\Desktop\student management app>npm init -y
Wrote to C:\Users\durga\OneDrive\Desktop\student management app\backend\package.json

{
  "name": "backend",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

32.Packages installed.

The screenshot shows the Visual Studio Code interface with the title bar "student management app". The Explorer sidebar on the left shows a project structure with "STUDENT MANAGEMENT APP" expanded, revealing "backend", "node_modules", ".env", "combined.log", "error.log", "package-lock.json", "package.json", and "server.js". The "server.js" file is selected in the Explorer. The Terminal tab at the top has "TERMINAL" selected. The terminal window displays the command "npm install express mongoose cors dotenv axios nodemon" being run in the "backend" directory, followed by the output showing package installation details. The status bar at the bottom right shows the date and time as "18 June, 2025".

```
C:\Users\durga\OneDrive\Desktop\student management app>cd backend
C:\Users\durga\OneDrive\Desktop\student management app>npm install express mongoose cors dotenv axios nodemon
added 125 packages, and audited 126 packages in 20s
22 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

C:\Users\durga\OneDrive\Desktop\student management app>npm install winston morgan
added 34 packages, and audited 160 packages in 6s
23 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

33.node.js server connection on port number 3000

The screenshot shows the Visual Studio Code interface with the title bar "student management app". The terminal window displays the command "node server.js" being run in the "backend" directory. The output shows MongoDB driver warnings about deprecated options like "useNewUrlParser" and "useUnifiedTopology", and it confirms that the server is running on port 3000 and connected to MongoDB.

```
C:\Users\durga\OneDrive\Desktop\student management app>cd backend
C:\Users\durga\OneDrive\Desktop\student management app>node server.js
(node:8292) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:8292) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is running on port 3000
Connected to MongoDB
```

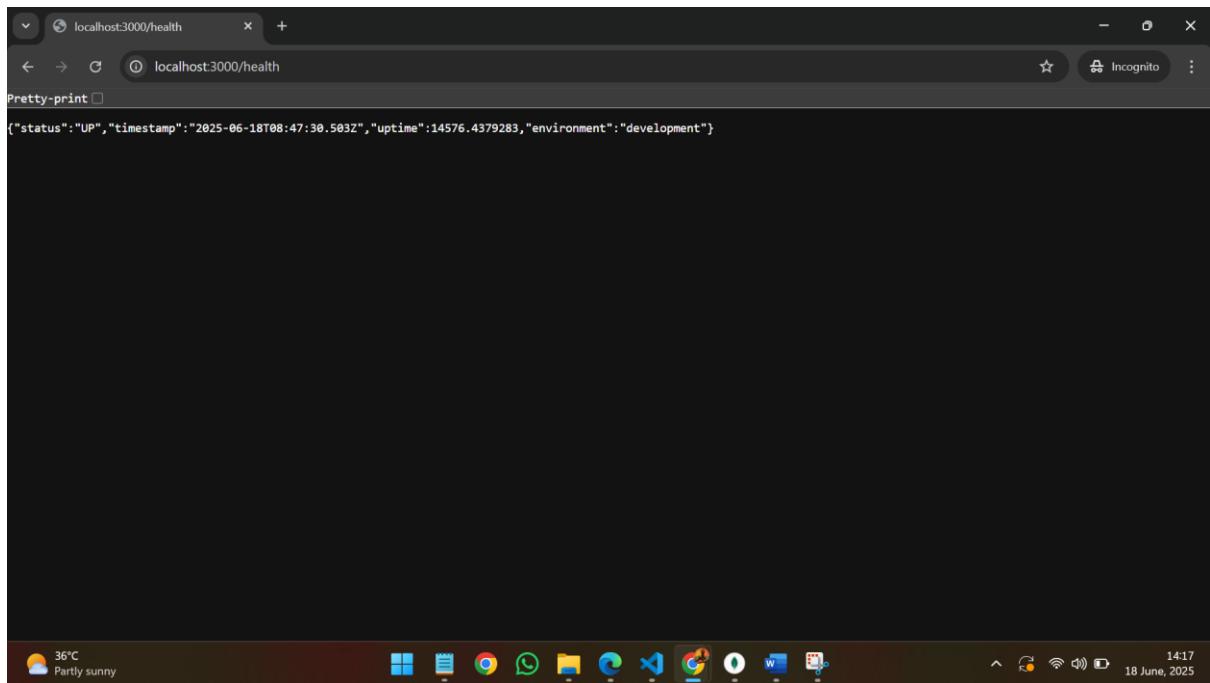
34.API calls

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "STUDENT MANAGEMENT APP" with "backend" selected. Inside "backend", there are "node_modules", ".env", "combined.log", "error.log", "package-lock.json", "package.json", and "server.js".
- Terminal:** The active tab shows log output from "server.js". The logs include several "info" messages indicating successful retrievals of students and dashboard statistics.
- Bottom Status Bar:** Shows the file path as "student management app", line 457, column 1, and the date/time as "18 June, 2025".

```
info: [object Object] {"timestamp": "2025-06-18T06:10:10.843Z"}  
info: Retrieved 7 students successfully {"timestamp": "2025-06-18T06:27:47.738Z"}  
GET /api/students 200 15.671 ms - 1779  
info: [object Object] {"timestamp": "2025-06-18T06:27:47.744Z"}  
info: Dashboard statistics retrieved successfully: {"activeCourses":2,"activeStudents":7,"courseCounts":[{"_id":"6852518725104da9d1cfa419","count":5}, {"_id":"685253fd25104da9d1cfa3fe","count":2}], "graduates":0, "successRate":0, "timestamp": "2025-06-18T06:27:47.782Z", "totalCourses":2, "totalStudents":7}  
GET /api/dashboard/stats 200 53.079 ms - 210  
info: [object Object] {"timestamp": "2025-06-18T06:27:47.786Z"}  
GET /api/students/685255e125104da9dicfa43a 200 8.067 ms - 252  
info: [object Object] {"timestamp": "2025-06-18T06:27:56.295Z"}  
GET /api/students/685255e125104da9dicfa43a 200 5.731 ms - 251  
info: [object Object] {"timestamp": "2025-06-18T06:28:31.229Z"}  
info: Retrieved 2 courses successfully {"timestamp": "2025-06-18T08:19:50.517Z"}  
GET /api/courses 304 23.421 ms - -  
info: [object Object] {"timestamp": "2025-06-18T08:19:50.527Z"}  
info: Retrieved 7 students successfully {"timestamp": "2025-06-18T08:19:50.544Z"}  
GET /api/students 200 10.267 ms - 1780  
info: [object Object] {"timestamp": "2025-06-18T08:19:50.547Z"}  
info: Dashboard statistics retrieved successfully: {"activeCourses":2,"activeStudents":7,"courseCounts":[{"_id":"685253fd25104da9d1cfa419","count":2}, {"_id":"6852518725104da9d1cfa3fe","count":5}], "graduates":0, "successRate":0, "timestamp": "2025-06-18T08:19:50.566Z", "totalCourses":2, "totalStudents":7}  
GET /api/dashboard/stats 200 29.140 ms - 210  
info: [object Object] {"timestamp": "2025-06-18T08:19:50.568Z"}  
info: Retrieved 7 students successfully {"timestamp": "2025-06-18T08:19:51.972Z"}  
GET /api/students 304 13.111 ms - -  
info: [object Object] {"timestamp": "2025-06-18T08:19:51.977Z"}  
info: Dashboard statistics retrieved successfully: {"activeCourses":2,"activeStudents":7,"courseCounts":[{"_id":"685253fd25104da9d1cfa419","count":2}], "graduates":0, "successRate":0, "timestamp": "2025-06-18T08:19:51.977Z"}  
GET /api/courses 304 5.908 ms - -  
info: [object Object] {"timestamp": "2025-06-18T08:30:12.558Z"}  
info: Retrieved 9 students successfully {"timestamp": "2025-06-18T08:30:12.568Z"}  
GET /api/students 200 5.225 ms - 2334  
info: [object Object] {"timestamp": "2025-06-18T08:30:12.570Z"}  
info: Dashboard statistics retrieved successfully: {"activeCourses":2,"activeStudents":9,"courseCounts":[{"_id":"6852518725104da9d1cfa3fe","count":5}, {"_id":"685253fd25104da9d1cfa419","count":2}], "graduates":0, "successRate":0, "timestamp": "2025-06-18T08:30:12.589Z", "totalCourses":2, "totalStudents":9}  
GET /api/dashboard/stats 200 25.847 ms - 255  
info: [object Object] {"timestamp": "2025-06-18T08:30:12.594Z"}  
info: Retrieved 9 students successfully {"timestamp": "2025-06-18T08:30:15.008Z"}  
GET /api/students 304 19.781 ms - -  
info: [object Object] {"timestamp": "2025-06-18T08:30:15.021Z"}  
info: Dashboard statistics retrieved successfully: {"activeCourses":2,"activeStudents":9,"courseCounts":[{"_id":"6852518725104da9d1cfa3fe","count":2}, {"_id":"685253fd25104da9d1cfa419","count":5}], "graduates":0, "successRate":0, "timestamp": "2025-06-18T08:30:15.055Z", "totalCourses":2, "totalStudents":9}  
GET /api/dashboard/stats 200 59.133 ms - 255  
info: [object Object] {"timestamp": "2025-06-18T08:30:15.060Z"}  
1412
```

35. end point “health”

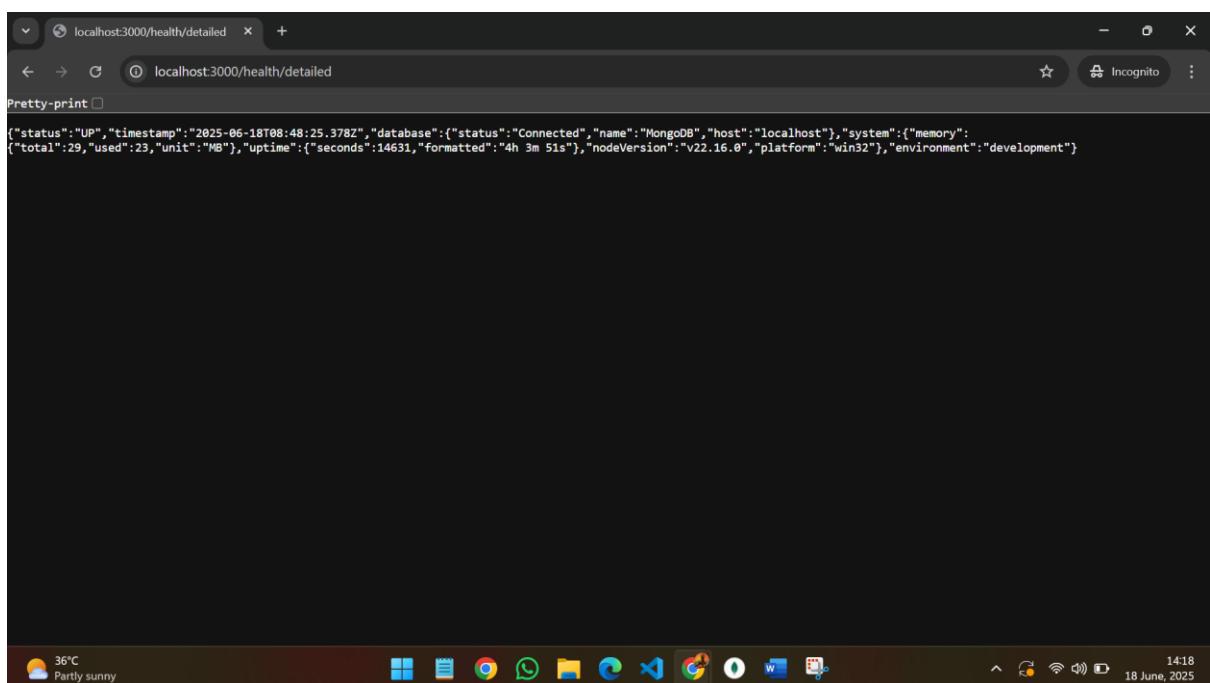


A screenshot of a Google Chrome browser window. The address bar shows "localhost:3000/health". The main content area displays a JSON object:

```
{"status": "UP", "timestamp": "2025-06-18T08:47:30.503Z", "uptime": 14576.4379283, "environment": "development"}
```

The browser interface includes standard controls like back, forward, and search, along with an "Incognito" tab and a menu icon. At the bottom, there's a taskbar with various pinned icons and system status indicators like battery level and date/time.

36.end point named “health detailed”



A screenshot of a Google Chrome browser window. The address bar shows "localhost:3000/health/detailed". The main content area displays a more detailed JSON object:

```
{"status": "UP", "timestamp": "2025-06-18T08:48:25.378Z", "database": {"status": "Connected", "name": "MongoDB", "host": "localhost"}, "system": {"memory": {"total": 29, "used": 23, "unit": "MB"}, "uptime": {"seconds": 14631, "formatted": "4h 3m 51s"}, "nodeVersion": "v22.16.0", "platform": "win32"}, "environment": "development"}
```

The browser interface and taskbar are identical to the previous screenshot, showing the same pinned icons and system status.