 Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: AWT (01CT1625)	Aim: Create a RESTful service with Node, Express and MongoDB.	
Experiment No: 6	Date:	Enrollment No: 92200133005

AIM: Backend: Using NodeJS, build a job search platform or e-commerce platform api to search and filter.

CODE:

```
const mongoose = require('mongoose');

const studentSchema = new mongoose.Schema({

  name: {

    type: String,

    required: true },

  rollNo: {

    type: String,

    required: true,

    unique: true },

  course: {

    type: String,

    required: true },

  semester: {

    type: Number,

    required: true },

  createdAt: {

    type: Date,

    default: Date.now


  }

});

module.exports = mongoose.model('Student', studentSchema);

const express = require('express');

const mongoose = require('mongoose');
```

 Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: AWT (01CT1625)	Aim: Create a RESTful service with Node, Express and MongoDB.	
Experiment No: 6	Date:	Enrollment No: 92200133005

```
const bodyParser = require('body-parser');

const app = express();

const PORT = process.env.PORT || 3000;

app.use(bodyParser.json());

// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/studentDB', {
  useNewUrlParser: true,
  useUnifiedTopology: true})
.then(() => console.log('Connected to MongoDB'))
.catch(err => console.error('Could not connect to MongoDB', err));

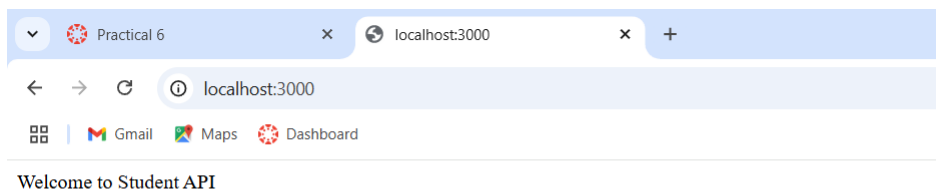
const studentRoutes = require('./routes/students');


app.use('/api/students', studentRoutes);

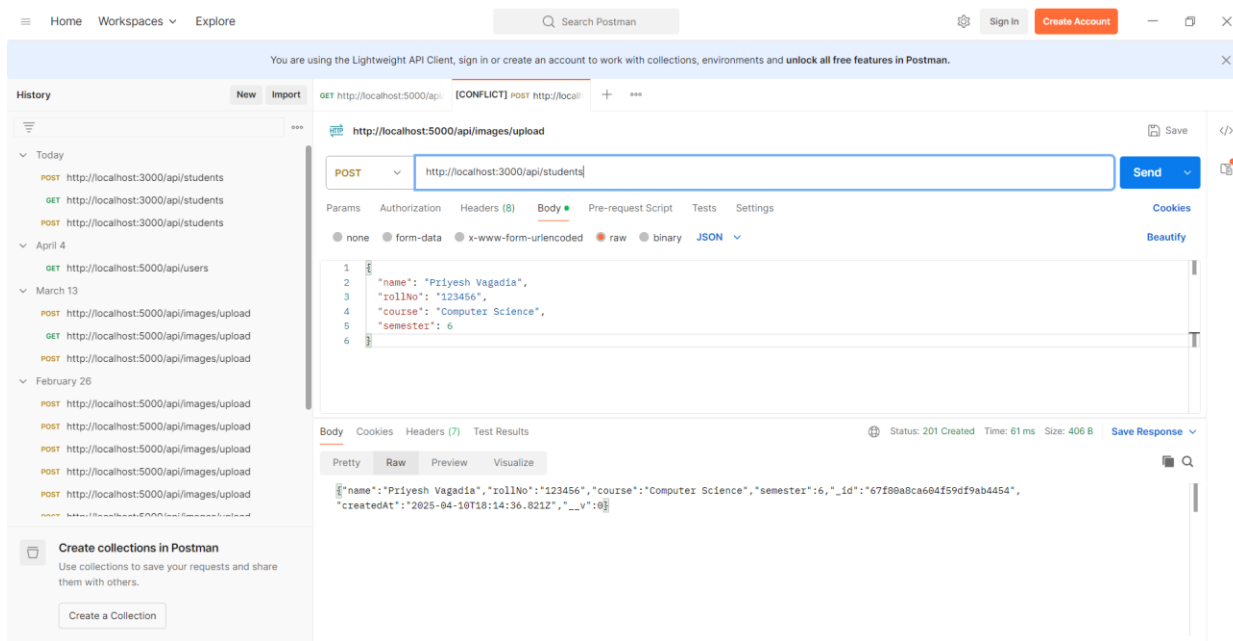
app.get('/', (req, res) => {
  res.send('Welcome to Student API');
});

// Start server
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```

OUTPUT:



 Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: AWT (01CT1625)	Aim: Create a RESTful service with Node, Express and MongoDB.	
Experiment No: 6	Date:	Enrollment No: 92200133005



The screenshot shows the Postman REST client interface. The top bar includes navigation links (Home, Workspaces, Explore), a search bar, and user options (Sign In, Create Account). A notification banner states: "You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman."

The left sidebar displays a "History" list of recent requests, categorized by date (Today, April 4, March 13, February 26). The main workspace shows a selected POST request to `http://localhost:3000/api/students`. The request body is a JSON object:

```

{
  "name": "Priyesh Vagadia",
  "rollNo": "123456",
  "course": "Computer Science",
  "semester": 6
}

```

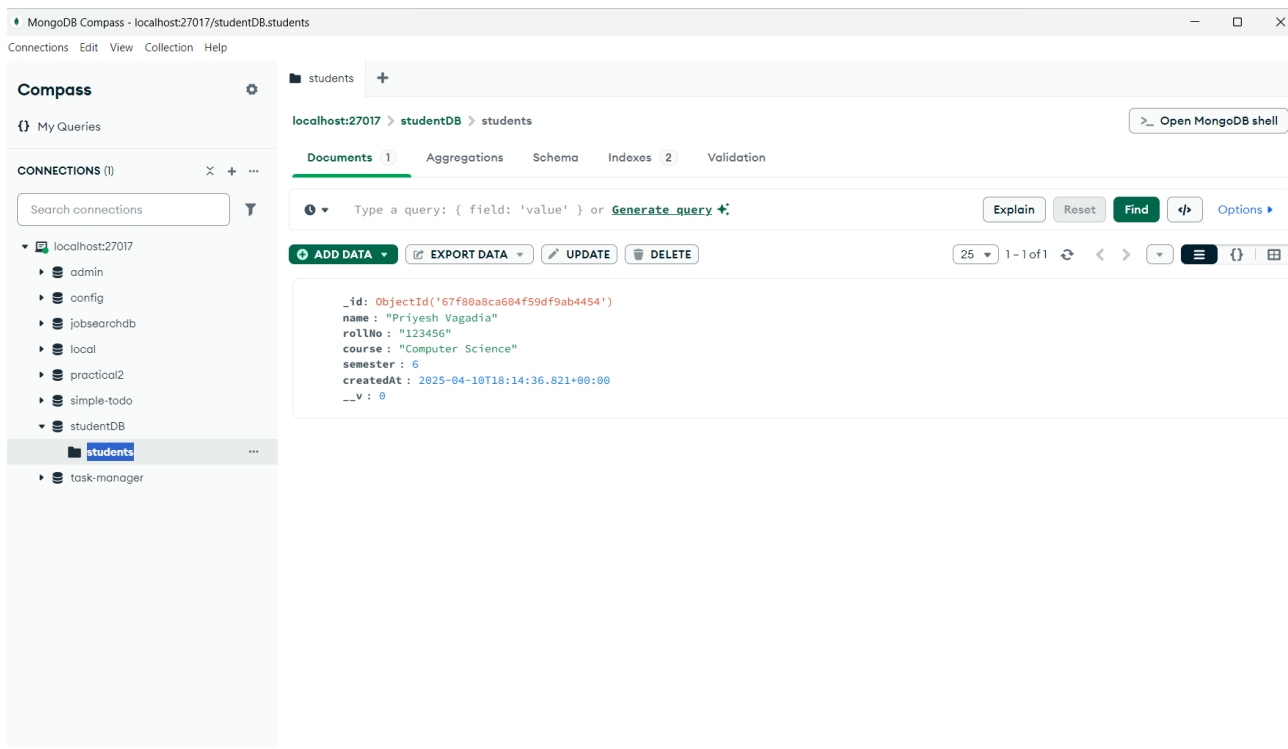
The bottom section shows the response in "Pretty" format, indicating a 201 status code (Created) with a response time of 61 ms and a size of 406 B. The response body is a JSON object:

```

{
  "name": "Priyesh Vagadia",
  "rollNo": "123456",
  "course": "Computer Science",
  "semester": 6,
  "_id": "67f80a8ca604f59df9ab4454",
  "createdAt": "2025-04-10T18:14:36.821Z",
  "__v": 0
}

```

At the bottom left, there is a "Create collections in Postman" button with a subtext: "Use collections to save your requests and share them with others."



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection: "MongoDB Compass - localhost:27017/studentDB.students". The left sidebar shows the "Connections" list with various databases and collections, including "students" which is currently selected.

The main workspace displays the "students" collection in the "Documents" tab. The top of the workspace shows the path: `localhost:27017 > studentDB > students`. Below this, there are tabs for "Documents", "Aggregations", "Schema", "Indexes", and "Validation".


The "Documents" tab shows a list of documents. The first document is displayed in a preview window:

```

{
  "_id": ObjectId('67f80a8ca604f59df9ab4454'),
  "name": "Priyesh Vagadia",
  "rollNo": "123456",
  "course": "Computer Science",
  "semester": 6,
  "createdAt": "2025-04-10T18:14:36.821+00:00",
  "__v": 0
}

```

At the bottom of the workspace, there are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". The bottom right corner shows pagination information: "25 1-1 of 1".

 Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: AWT (01CT1625)	Aim: Create a RESTful service with Node, Express and MongoDB.	
Experiment No: 6	Date:	Enrollment No: 92200133005

GET http://localhost:5000/api/
[CONFLICT] PUT http://localhc
+ ...

HTTP
http://localhost:5000/api/images/upload

PUT
▼
http://localhost:3000/api/students/STUDENT_ID

Params
Authorization
Headers (8)
Body ●
Pre-request Script
Tests
Settings

none
form-data
x-www-form-urlencoded
raw ●
binary
JSON ▼

```

1
2
3
"course": "Information Technology"

```