

Day-3 outputs

PostMapping:

The screenshot shows a Postman interface for a POST request to `http://localhost:8088/student/add`. The request body is a JSON object: `{ "id": 18, "name": "pk", "age": 23, "salary": 27880 }`. The response is a 201 Created status with a 76 ms response time and 218 B of data. The response body is a JSON object: `{ "id": "18", "name": "pk", "age": 23, "salary": 27880.0 }`.

```
1 {
2   "id": 18,
3   "name": "pk",
4   "age": 23,
5   "salary": 27880
6 }
```

```
1 {
2   "id": "18",
3   "name": "pk",
4   "age": 23,
5   "salary": 27880.0
6 }
```

PutMapping:

The screenshot shows a Postman interface for a PUT request to `http://localhost:8088/student/update/45`. The request body is a JSON object: `{ "id": 45, "name": "ramesh", "age": 45, "salary": 80000 }`. The response is a 200 OK status with a 67 ms response time and 217 B of data. The response body is a JSON object: `{ "id": "45", "name": "ramesh", "age": 45, "salary": 80000.0 }`.

```
1 {
2   "id": 45,
3   "name": "ramesh",
4   "age": 45,
5   "salary": 80000
6 }
```

```
1 {
2   "id": "45",
3   "name": "ramesh",
4   "age": 45,
5   "salary": 80000.0
6 }
```

GetMapping:

GET http://localhost:8088/student/all

Send

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

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

```
1 [
2   {
3     "id": "18",
4     "name": "pk",
5     "age": 23,
6     "salary": 27880.0
7   }
8 ]
```

Body Cookies Headers (5) Test Results 200 OK 84 ms 321 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "18",
4     "name": "pk",
5     "age": 23,
6     "salary": 27880.0
7   }
8 ]
```

Delete:

DEL http://localhost:8088/student/delete/25

Send

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

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

```
1 [
2   {
3     "id": "25",
4     "name": "suresh",
5     "age": 55,
6     "salary": 110000
7   }
8 ]
```

Body Cookies Headers (3) Test Results 204 No Content 57 ms 112 B Save Response

Pretty Raw Preview Visualize Text

```
1
```

Database:

The screenshot displays the MongoDB Atlas web interface. The browser address bar shows the URL: `cloud.mongodb.com/v2/63033213da931339fad9b77d#/metrics/replicaSet/652bbc6717baef65ad940680/explorer/studentDetails/studentdb/find`. The Atlas logo is in the top left, and the user's name 'Pavankalyan...' is in the top right. The main navigation bar includes 'Project 0', 'Data Services', 'App Services', and 'Charts'. The left sidebar shows a tree view with 'Overview', 'DEPLOYMENT', 'Database' (selected), 'Data Lake', 'SERVICES', 'Device Sync', 'Triggers', 'Data API', 'Data Federation', 'Search', 'Stream Processing', and 'SECURITY'. The 'Database' section is expanded, showing a list of databases: 'sample_airbnb', 'sample_analytics', 'sample_geospatial', 'sample_guides', 'sample_restaurants', 'sample_supplies', 'sample_training', 'sample_weatherdata', and 'studentDetails' (highlighted in green). The main panel shows the 'Find' tab with a query filter input: 'Type a query: { field: 'value' }'. Below the input, it says 'QUERY RESULTS: 1-3 OF 3'. A single document is displayed with the following fields: `_id: "18"`, `name: "pk"`, `age: 23`, `salary: 27880`, and `_class: "com.example.demo.Student"`. At the bottom, the system status is 'All Good'.