



CSE471 : System Analysis and Design
Lab Assignment on API Development and Postman

Project Title : AbroadEase
Group Number : 05
Backend Framework : MERN

Submitted by : Tasnim Rahman Moumita
Student ID : 22301689
Section: 02

Date : 08.08.2025

Instruction:

1. You have to develop all the REST APIs/Endpoints of two features (which interfaces you have submitted in the Figma assignment) of your project using your preferred backend framework. Database should be connected.
2. This is an individual submission. The developed APIs can not be duplicated among group members.
3. Add a code snippet for each API.
4. Run a backend server on a specific port. The port number should be the **last four digits of your student id**.

Test your developed API in Postman. Add a screenshot for each API.

APIs of Feature - 01

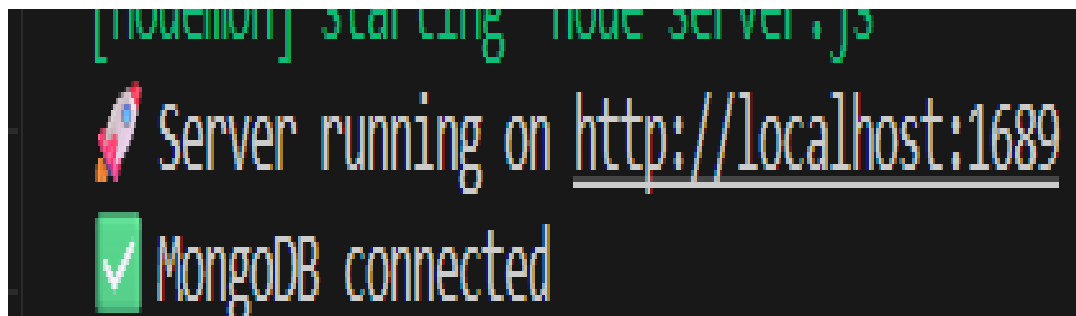
Module 2: Post guidelines and filtering:

Feature-4:

Admin will be able to write posts regarding SOP instructions.

* Last 4 digits of my student ID- 1689

Screenshots:



HTTP <http://localhost:1689> Save Share

GET <http://localhost:1689> Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (8) Test Results 200 OK • 18 ms • 290 B •

HTML Preview Visualize

```
1 Hello! Server is working fine.
```

```
JS server.js > ...
1 // server.js
2 require('dotenv').config();
3 const express = require('express');
4 const mongoose = require('mongoose');
5 const cors = require('cors');
6 const morgan = require('morgan');
7
8 const app = express();
9 const PORT = process.env.PORT || 1689;
10 const MONGO_URI = process.env.MONGODB_URI;
11
12 // Middleware
13 app.use(express.json());
14 app.use(cors());
15 app.use(morgan('dev'));
16
17 // DB connect
18 mongoose
19   .connect(MONGO_URI, { autoIndex: true })
20   .then(() => console.log('✅ MongoDB connected'))
21   .catch((err) => {
22     console.error('❌ MongoDB connection error:', err.message);
23     process.exit(1);
24   });
25
```

```

JS server.js > ...
37
38 const SopPost = mongoose.model('SopPost', SopPostSchema);
39
40 // Health (optional, helps you show server is up)
41 app.get('/api/health', (req, res) => {
42   res.json({ status: 'ok', time: new Date().toISOString() });
43 });
44
45 // Feature: Admin writes SOP instruction posts
46 // Endpoint: POST /api/sop-posts
47 app.post('/api/sop-posts', async (req, res) => {
48   try {
49     const { title, country, degreeLevel = 'Other', tags = [], content } = req.body;
50
51     // Minimal validation (fast and clear errors)
52     if (!title || title.length < 5) {
53       return res.status(400).json({ error: 'Title is required and must be at least 5 characters.' });
54     }
55     if (!country || country.length < 2) {
56       return res.status(400).json({ error: 'Country is required and must be at least 2 characters.' });
57     }
58     if (!content || content.length < 50) {
59       return res.status(400).json({ error: 'Content is required and must be at least 50 characters.' });
60     }
61

```

```

JS server.js > ...
47 app.post('/api/sop-posts', async (req, res) => {
60   }
61
62   const doc = await SopPost.create({ title, country, degreeLevel, tags, content });
63   return res.status(201).json({ message: 'SOP post created', data: doc });
64 } catch (err) {
65   console.error('Create SOP error:', err);
66   return res.status(500).json({ error: 'Internal server error' });
67 }
68 });
69
70 // Not found fallback
71 app.use((req, res) => {
72   res.status(404).json({ error: 'Route not found' });
73 });
74
75 // Start server on 1689
76 app.listen(PORT, () => {
77   console.log(`🚀 Server running on http://localhost:${PORT}`);
78 });
79
80

```

HTTP <http://localhost:1689/api/sop-posts> Save Share

POST <http://localhost:1689/api/sop-posts> Send

Params Authorization **Headers (8)** Body Scripts Settings Cookies

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Postman-Token	<calculated when request is sent>				
<input checked="" type="checkbox"/>	Content-Type	application/json				
<input checked="" type="checkbox"/>	Content-Length	<calculated when request is sent>				
<input checked="" type="checkbox"/>	Host	<calculated when request is sent>				
<input checked="" type="checkbox"/>	User-Agent	PostmanRuntime/7.45.0				
<input checked="" type="checkbox"/>	Accept	*/*				

Body Cookies Headers (8) Test Results 201 Created 305 ms 971 B 🌐 ...

{ } JSON Preview Visualize 🔍 📄 🔗

```

1  {
2    "message": "SOP post created",
3    "data": {
4      "title": "SOP Instructions for Masters in Computer Science",
5      "country": "Canada",
6      "degreeLevel": "Masters",
7      "tags": [
8        "SOP",
9        "Masters",
10       "Canada"
11     ],

```

Body Cookies Headers (8) Test Results 201 Created 305 ms 971 B 🌐 ...

{ } JSON Preview Visualize 🔍 📄 🔗

```

10    "Canada"
11  ],
12  "content": "Begin with a clear motivation statement linking your background to your research
13             interests. Describe two to three significant projects with results and your contributions.
14             Explain why this program fits your goals with specifics (faculty, labs, courses). State
15             short- and long-term goals and how the program enables them. Keep it structured,
16             authentic, and concise, then proofread carefully.",
17  "_id": "6896434c098c9e6d080291a8",
18  "createdAt": "2025-08-08T18:34:52.138Z",
19  "updatedAt": "2025-08-08T18:34:52.138Z",
20  "__v": 0

```

HTTP <http://localhost:1689/api/sop-posts> Save Share

POST <http://localhost:1689/api/sop-posts> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
2  "title": "SOP Instructions for Masters in Computer Science",
3  "country": "Canada",
4  "degreeLevel": "Masters",
5  "tags": ["SOP", "Masters", "Canada"],
6  "content": "Begin with a clear motivation statement linking your background to your research
interests. Describe two to three significant projects with results and your contributions. Explain
why this program fits your goals with specifics (faculty, labs, courses). State short- and long-term
goals and how the program enables them. Keep it structured, authentic, and concise, then proofread
carefully."
7  }
```

Body Cookies Headers (8) Test Results 201 Created • 305 ms • 971 B

{ } JSON Preview Visualize

```
10  |   "Canada"
11  |   ],
12  |   "content": "Begin with a clear motivation statement linking your background to your research
interests. Describe two to three significant projects with results and your contributions. Explain
why this program fits your goals with specifics (faculty, labs, courses). State short- and long-term
goals and how the program enables them. Keep it structured, authentic, and concise, then proofread
carefully.",
13  |   "_id": "6896434c098c9e6d080291a8",
14  |   "createdAt": "2025-08-08T18:34:52.138Z",
15  |   "updatedAt": "2025-08-08T18:34:52.138Z",
16  |   "__v": 0
17  | }
18  | }
```

HTTP <http://localhost:1689/api/sop-posts> Save Share

POST <http://localhost:1689/api/sop-posts> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
2  "title": "SOP Instructions for Masters in Computer Science",
3  "country": "Canada",
4  "degreeLevel": "Masters",
5  "tags": ["SOP", "Masters", "Canada"],
6  "content": "Begin with a clear motivation statement linking your background to your research
interests. Describe two to three significant projects with results and your contributions. Explain
why this program fits your goals with specifics (faculty, labs, courses). State short- and long-term
goals and how the program enables them. Keep it structured, authentic, and concise, then proofread
carefully."
7  }
```

Body Cookies Headers (8) Test Results 201 Created • 305 ms • 971 B

{ } JSON Preview Visualize

Root / data

title	SOP Instructions for Masters in Computer Science
country	Canada
degreeLevel	Masters
tags	[3]
content	Begin with a clear motivation statement linking your background to your research interests. Describe two to three significant projects with results and your contributions. Explain why this program fits your goals with specifics (faculty, labs, courses). State short- and long-term goals and how the program enables them. Keep it structured, authentic, and concise, then proofread carefully.

Postbot Runner Start Proxy Cookies Vault Trash

APIs of Feature - 02

Module 3: Housing: Feature-2:

Detailed Property Information: Each listing includes property photos, descriptions, rental terms, and amenities for better decision-making.

```
routes > JS propertyRoutes.js > ...
 1  const express = require('express');
 2  const router = express.Router();
 3  const {
 4    getAllProperties,
 5    getPropertyById,
 6    createProperty,
 7    updateProperty,
 8    deleteProperty
 9  } = require('../controllers/propertyController');
10
11  // Get all properties
12  router.get('/', getAllProperties);
13
14  // Get property by ID
15  router.get('/:id', getPropertyById);
16
17  // Create new property
18  router.post('/', createProperty);
19
20  // Update property
21  router.put('/:id', updateProperty);
22
23  // Delete property
24  router.delete('/:id', deleteProperty);
25
26  module.exports = router;
```

```

JS server.js  {} package-lock.json  JS propertyController.js X  JS Property.js  JS propertyRoutes.js
controllers > JS propertyController.js > updateProperty > updateProperty
14 exports.createProperty = async (req, res) => {
24   };
25
26   // Create new property
27   exports.createProperty = async (req, res) => {
28     try {
29       const property = new Property(req.body);
30       const savedProperty = await property.save();
31       res.status(201).json(savedProperty);
32     } catch (err) {
33       res.status(400).json({ error: 'Failed to create property' });
34     }
35   };
36
37   // Update property
38   exports.updateProperty = async (req, res) => {
39     try {
40       const property = await Property.findByIdAndUpdate(
41         req.params.id,
42         req.body,
43         { new: true }
44       );
45       if (!property) {
46         return res.status(404).json({ error: 'Property not found' });
47       }
48       res.json(property);
49     } catch (err) {
50       res.status(400).json({ error: 'Failed to update property' });
51     }
52   };
53

```

```

54 // Delete property
55 exports.deleteProperty = async (req, res) => {
56   try {
57     const property = await Property.findByIdAndDelete(req.params.id);
58     if (!property) {
59       return res.status(404).json({ error: 'Property not found' });
60     }
61     res.json({ message: 'Property deleted successfully' });
62   } catch (err) {
63     res.status(500).json({ error: 'Failed to delete property' });
64   }
65 };
66
67
68 exports.createProperty = async (req, res) => {
69   try {
70     const newProperty = new Property(req.body);
71     const savedProperty = await newProperty.save();
72     res.status(201).json({
73       message: 'Property added successfully',
74       propertyId: savedProperty._id
75     });
76   } catch (error) {
77     console.error('✖ Error in createProperty:', error);
78     res.status(500).json({ error: error.message });
79   }
80 };
81

```


HTTP <http://localhost:1689/api/properties> Save Share

POST <http://localhost:1689/api/properties> Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "title": "Test Apartment",
3   "description": "A spacious flat near East London.",
4   "rentalTerms": "12-month lease",
5   "amenities": ["WiFi", "Lift", "CCTV"],
6   "photos": [
7     "http://example.com/photo1.jpg",
8     "http://example.com/photo2.jpg"
9   ]
10 }
```

Body Cookies Headers (8) Test Results 201 Created • 60 ms • 530 B

{ JSON Preview Visualize

```
4   "rentalTerms": "12-month lease",
5   "amenities": [
6     "WiFi",
7     "Lift",
8     "CCTV"
9   ],
10  "photos": [
11    "http://example.com/photo1.jpg",
12    "http://example.com/photo2.jpg"
13  ],
14  "_id": "6896187977349d9cff74a700",
15  "__v": 0
16 }
```

Postbot Runner Start Proxy Cookies Vault Trash

HTTP

http://localhost:1689/api/properties

Save

Share

GET

http://localhost:1689/api/properties

Send

Params

Authorization

Headers (8)

Body

Scripts

Settings

Cookies

Beautiful

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1  {
2    "title": "Test Apartment",
3    "description": "A spacious flat near East London.",
4    "rentalTerms": "12-month lease",
5    "amenities": ["WiFi", "Lift", "CCTV"],
6    "photos": [
7      "http://example.com/photo1.jpg",
8      "http://example.com/photo2.jpg"
9    ]
10 }
```

Body

Cookies

Headers (8)

Test Results

201 Created

60 ms

530 B

{ }

JSON

Preview

Visualize

title	Test Apartment
description	A spacious flat near Dhanmondi.
rentalTerms	12-month lease
> amenities	[3]
> photos	[2]
_id	6896187977349d9cff74a700
_v	0