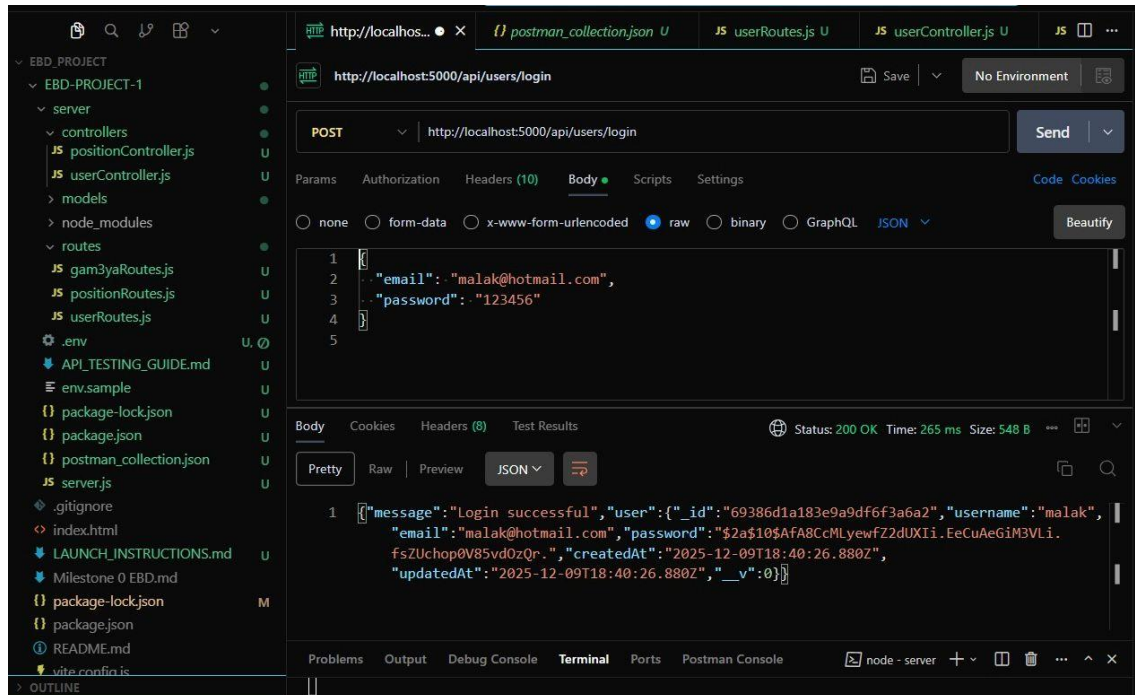
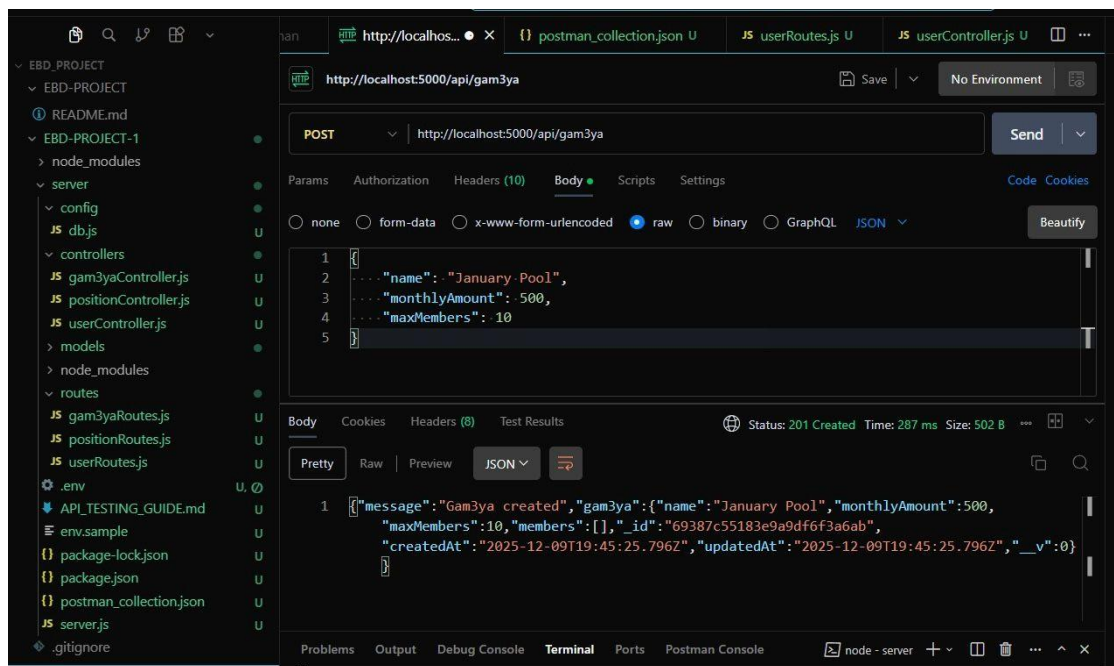


API testing using postman screenshots:

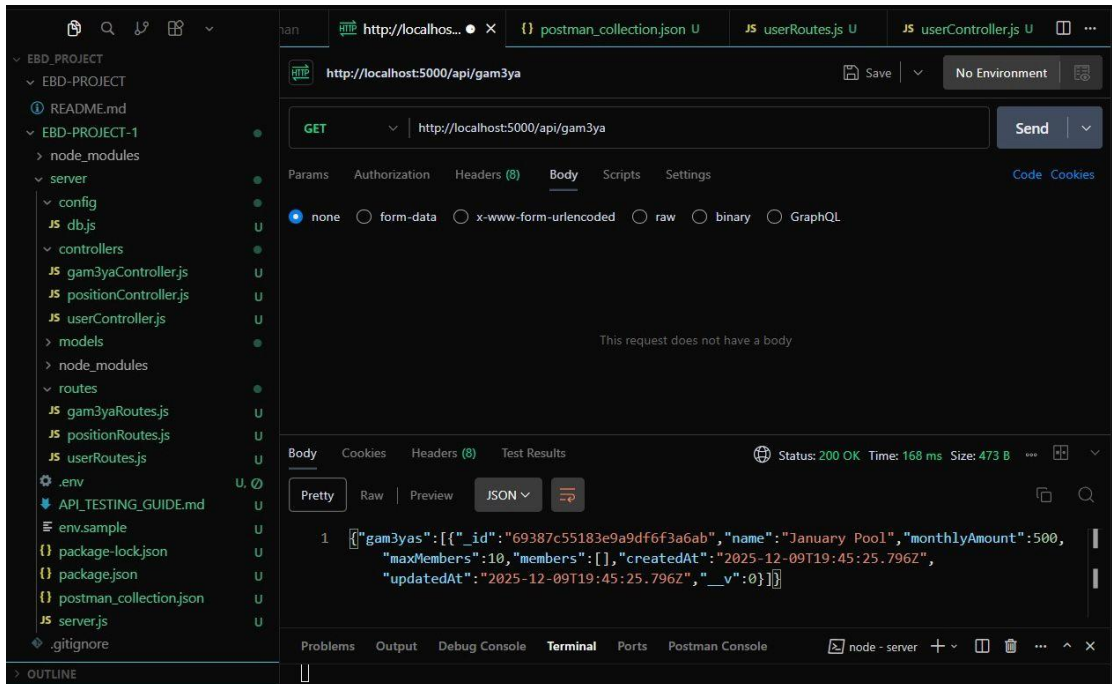
1. Login API



2. Create Gam3eya API



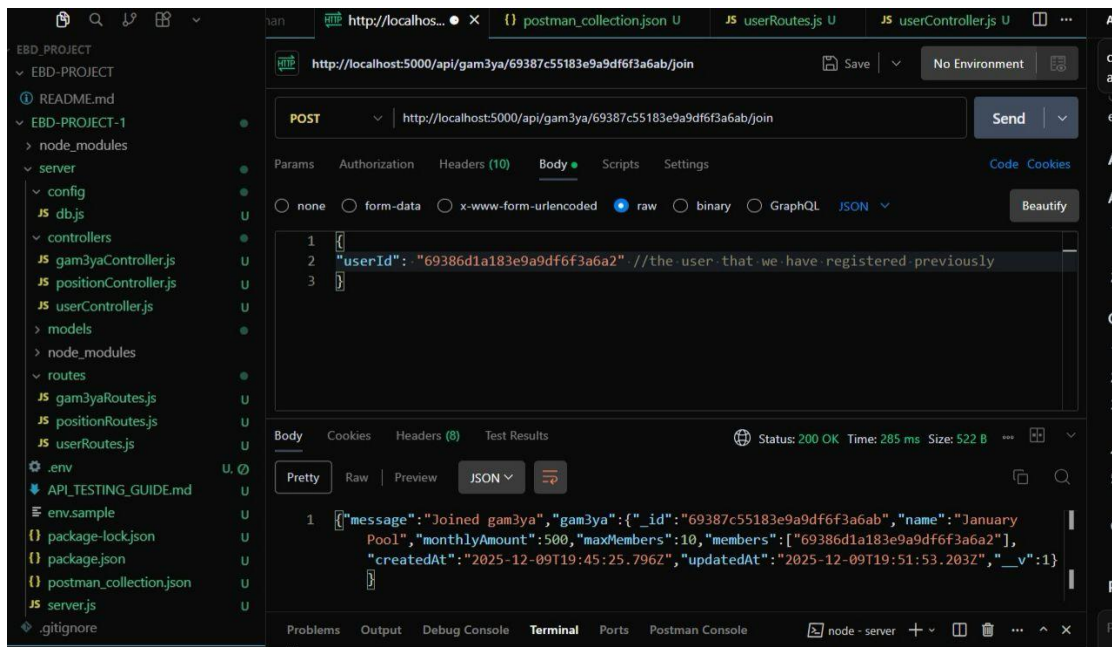
3. Get All Gam3eyas API



The screenshot shows a Postman interface with a GET request to `http://localhost:5000/api/gam3ya`. The response is a JSON array of gam3ya objects. The response status is 200 OK, with a time of 168 ms and a size of 473 B.

```
1 [{"_id": "69387c55183e9a9df6f3a6ab", "name": "January Pool", "monthlyAmount": 500, "maxMembers": 10, "members": [], "createdAt": "2025-12-09T19:45:25.796Z", "updatedAt": "2025-12-09T19:45:25.796Z", "__v": 0}]
```

4. Join Gam3eya API



The screenshot shows a Postman interface with a POST request to `http://localhost:5000/api/gam3ya/69387c55183e9a9df6f3a6ab/join`. The request body is a JSON object with a `userId` field. The response is a JSON object with a `message` field and gam3ya details. The response status is 200 OK, with a time of 285 ms and a size of 522 B.

```
1 {"message": "Joined gam3ya", "gam3ya": {"_id": "69387c55183e9a9df6f3a6ab", "name": "January Pool", "monthlyAmount": 500, "maxMembers": 10, "members": ["69386d1a183e9a9df6f3a6a2"], "createdAt": "2025-12-09T19:45:25.796Z", "updatedAt": "2025-12-09T19:51:53.203Z", "__v": 1}}
```

5. Get Gam3eya By ID API

A screenshot of the Postman application interface. The left sidebar shows a file explorer for a project named 'EBD_PROJECT'. The main workspace displays a GET request to the endpoint `http://localhost:5000/api/gam3ya/69387c55183e9a9df6f3a6ab`. The 'Body' tab is selected, showing a JSON response with details for a Gam3eya entry, including its ID, name, monthly amount, and members. The status bar at the bottom indicates a successful 200 OK response.

```
GET http://localhost:5000/api/gam3ya/69387c55183e9a9df6f3a6ab
```

Body

```
1 [{"gam3ya":{"_id":"69387c55183e9a9df6f3a6ab","name":"January Pool","monthlyAmount":500,"maxMembers":10,"members":[{"_id":"69386d1a183e9a9df6f3a6a2","username":"malak","email":"malak@hotmail.com"}],"createdAt":"2025-12-09T19:45:25.796Z","updatedAt":"2025-12-09T19:51:53.203Z","__v":1}}]
```

6. Update Gam3eya API

A screenshot of the Postman application interface showing a PUT request to the same endpoint as in the previous screenshot. The 'Body' tab is selected, showing a JSON request to update the Gam3eya entry. The status bar at the bottom indicates a successful 200 OK response.

```
PUT http://localhost:5000/api/gam3ya/69387c55183e9a9df6f3a6ab
```

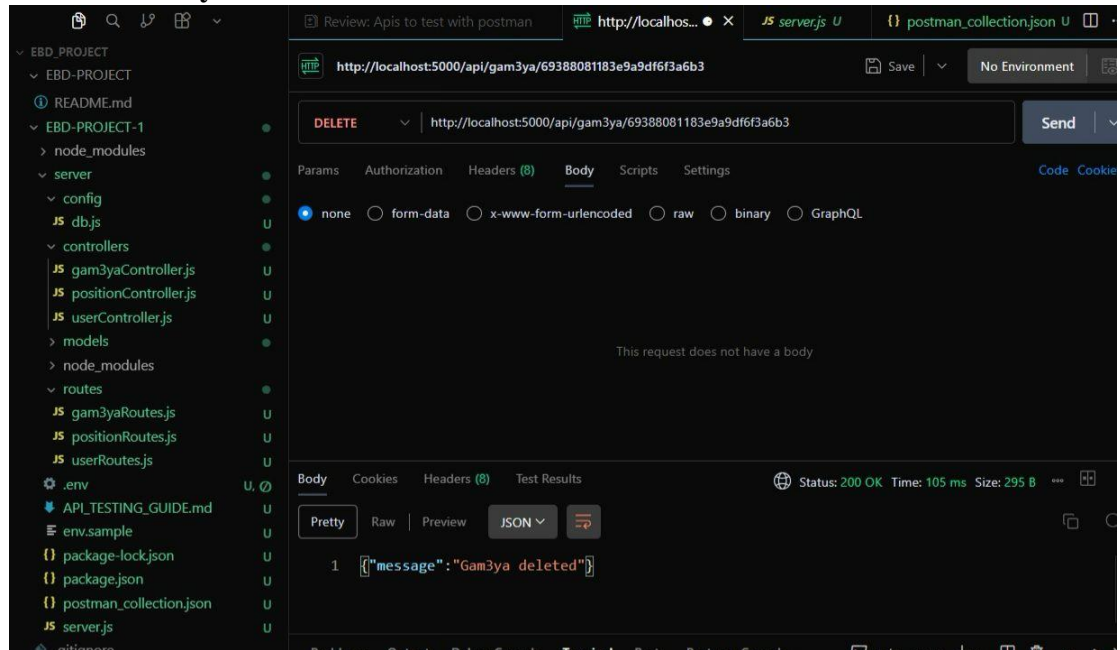
Body

```
1 {"name": "JanuaryPoolUpdated",
2  "monthlyAmount": 600
3 }
4
```

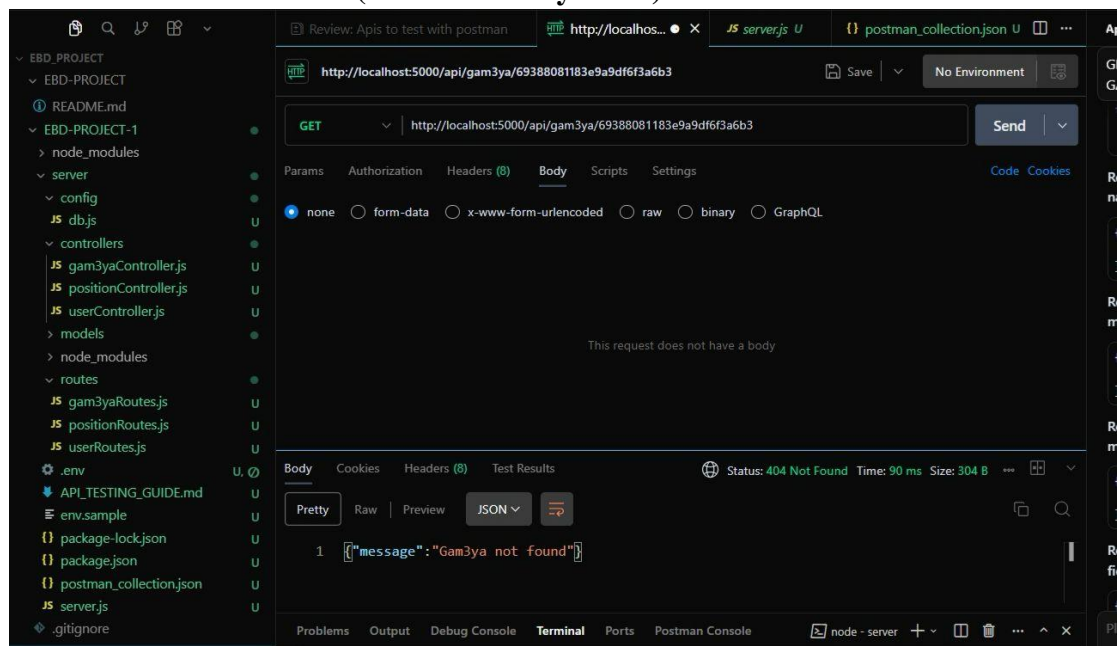
Body

```
1 [{"message":"Gam3ya updated","gam3ya":{"_id":"69387c55183e9a9df6f3a6ab","name":"JanuaryPoolUpdated","monthlyAmount":600,"maxMembers":10,"members":[{"_id":"69386d1a183e9a9df6f3a6a2","username":"malak","email":"malak@hotmail.com"}],"createdAt":"2025-12-09T19:45:25.796Z","updatedAt":"2025-12-09T19:59:26.900Z","__v":1}}]
```

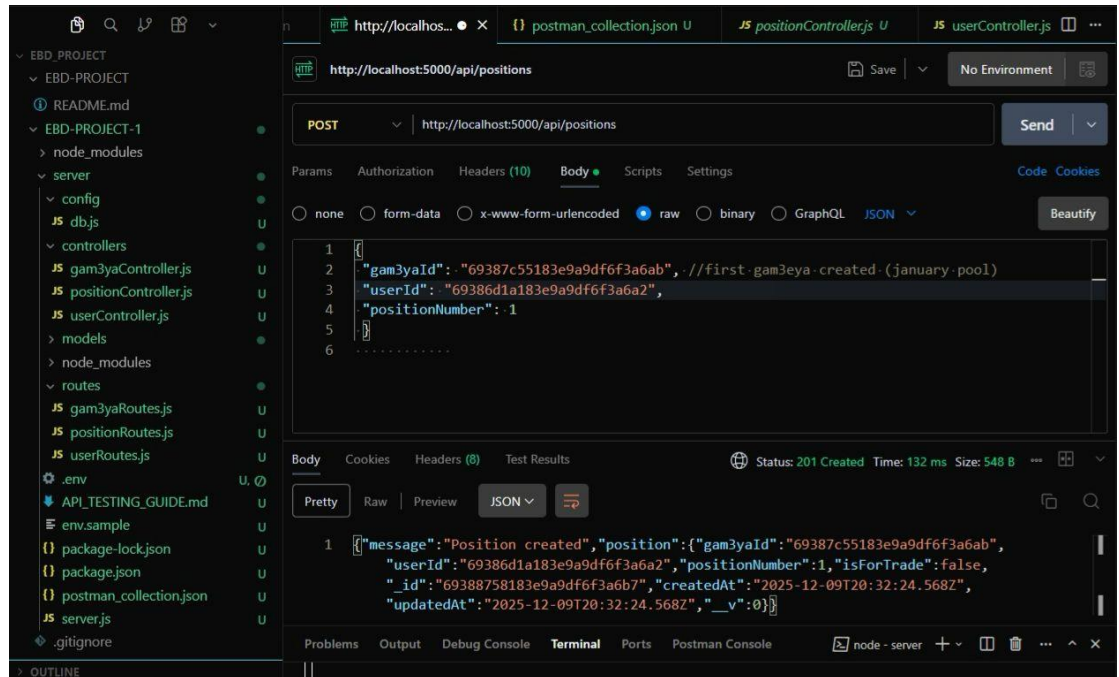
7. Delete Gam3eya API



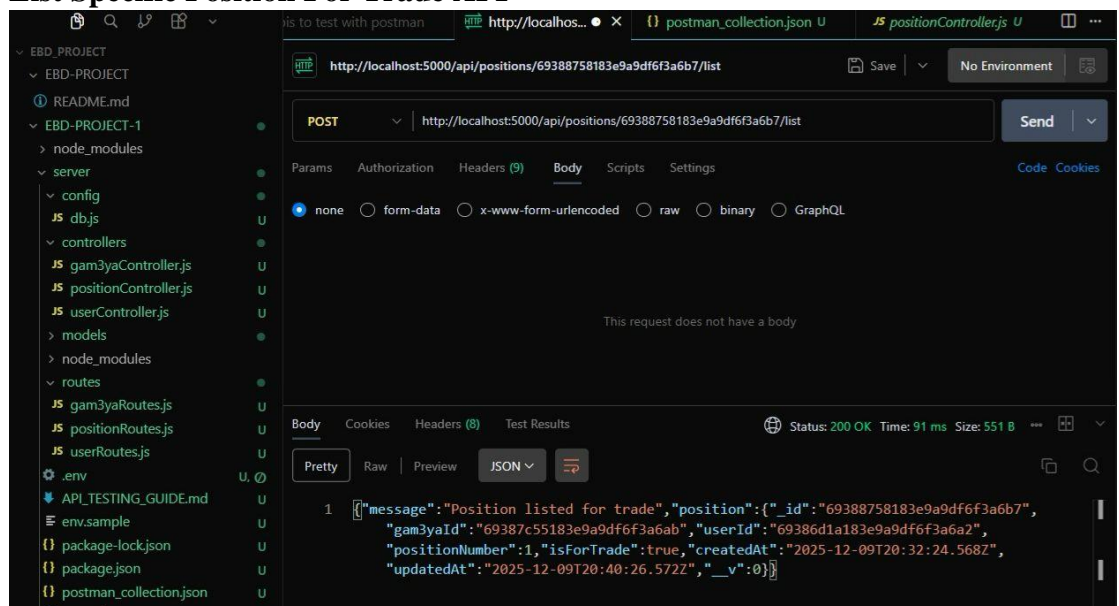
Confirmation of deletion (Delete Gam3eya API)



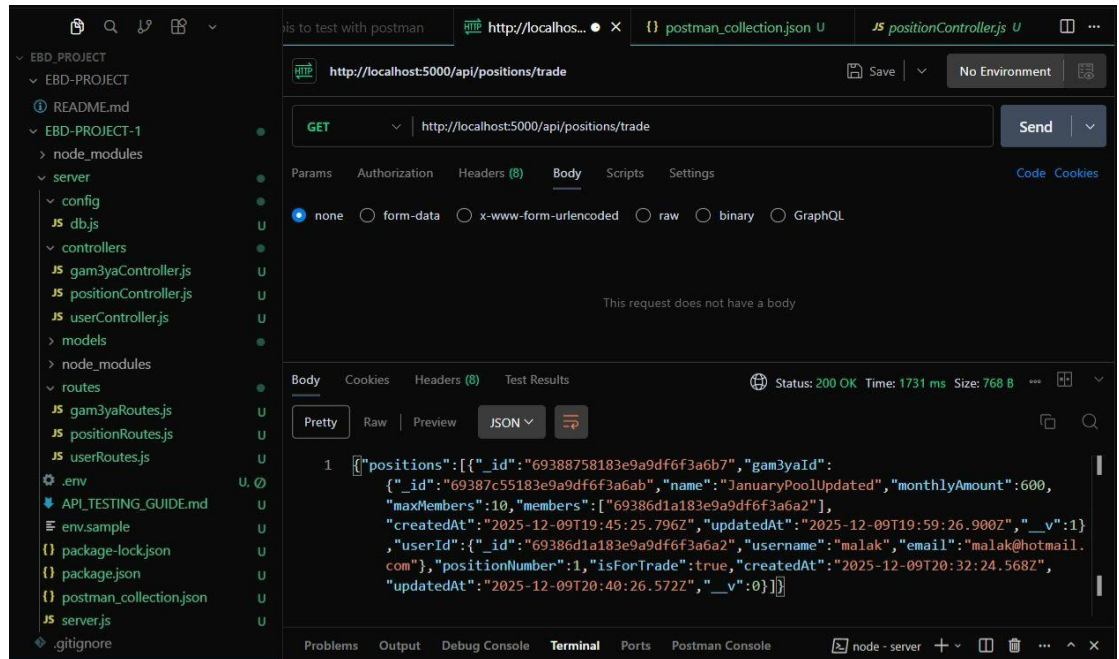
8. Create Position API



9. List Specific Position For Trade API



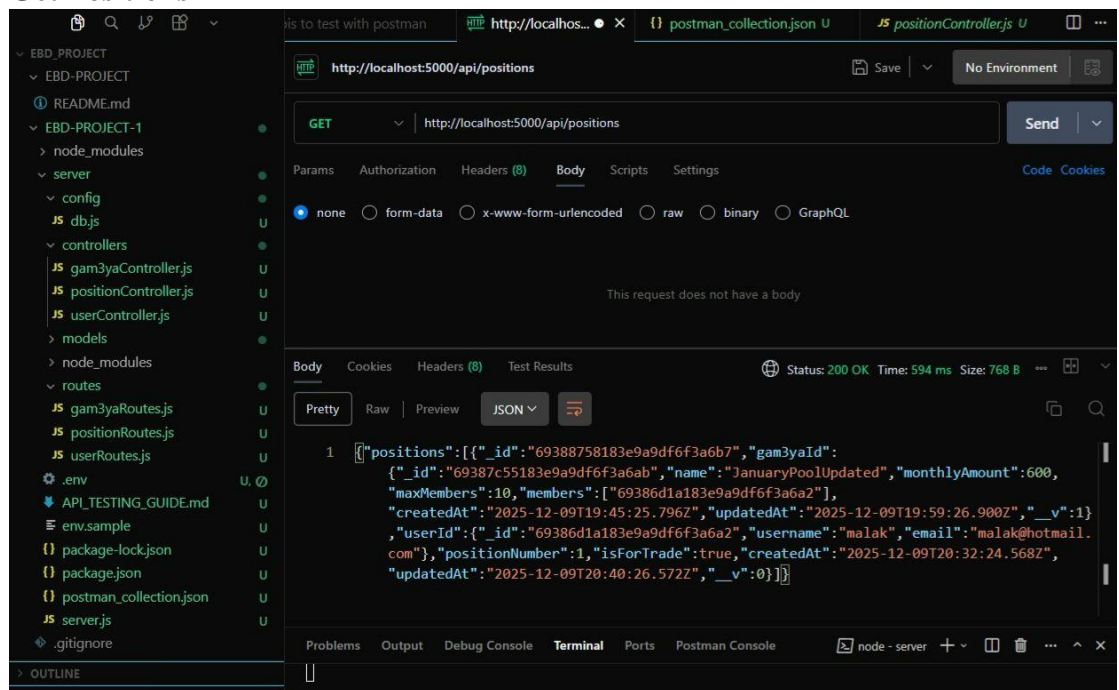
10. Get All Positions Listed For Trade API



The screenshot shows a Postman interface with a GET request to `http://localhost:5000/api/positions/trade`. The response is a JSON array containing one position object. The status is 200 OK, and the time taken is 1731 ms.

```
1 [{"_id": "69388758183e9a9df6f3a6b7", "gam3yaId": {"_id": "69387c55183e9a9df6f3a6ab", "name": "JanuaryPoolUpdated", "monthlyAmount": 600, "maxMembers": 10, "members": [{"_id": "69386d1a183e9a9df6f3a6a2"}, {"_id": "69386d1a183e9a9df6f3a6a2"}], "createdAt": "2025-12-09T19:45:25.796Z", "updatedAt": "2025-12-09T19:59:26.900Z", "__v": 1}, "userId": {"_id": "69386d1a183e9a9df6f3a6a2", "username": "malak", "email": "malak@hotmail.com"}, "positionNumber": 1, "isForTrade": true, "createdAt": "2025-12-09T20:32:24.568Z", "updatedAt": "2025-12-09T20:40:26.572Z", "__v": 0}]]
```

11. Get Positions API



The screenshot shows a Postman interface with a GET request to `http://localhost:5000/api/positions`. The response is a JSON array containing one position object. The status is 200 OK, and the time taken is 594 ms.

```
1 [{"_id": "69388758183e9a9df6f3a6b7", "gam3yaId": {"_id": "69387c55183e9a9df6f3a6ab", "name": "JanuaryPoolUpdated", "monthlyAmount": 600, "maxMembers": 10, "members": [{"_id": "69386d1a183e9a9df6f3a6a2"}, {"_id": "69386d1a183e9a9df6f3a6a2"}], "createdAt": "2025-12-09T19:45:25.796Z", "updatedAt": "2025-12-09T19:59:26.900Z", "__v": 1}, "userId": {"_id": "69386d1a183e9a9df6f3a6a2", "username": "malak", "email": "malak@hotmail.com"}, "positionNumber": 1, "isForTrade": true, "createdAt": "2025-12-09T20:32:24.568Z", "updatedAt": "2025-12-09T20:40:26.572Z", "__v": 0}]]
```

12. Get Positions By ID API

The screenshot shows a Postman interface with a GET request to the endpoint `http://localhost:5000/api/positions/69388758183e9a9df6f3a6b7`. The request is successful, returning a 200 OK status. The response body is a JSON object representing a position.

```
1 {
  "position": {
    "_id": "69388758183e9a9df6f3a6b7",
    "gam3yaId": {
      "_id": "69387c55183e9a9df6f3a6ab",
      "name": "JanuaryPoolUpdated",
      "monthlyAmount": 600,
      "maxMembers": 10,
      "members": [
        {
          "_id": "69386d1a183e9a9df6f3a6a2",
          "createdAt": "2025-12-09T19:45:25.796Z",
          "updatedAt": "2025-12-09T19:59:26.900Z",
          "v": 1
        },
        {
          "_id": "69386d1a183e9a9df6f3a6a2",
          "username": "malak",
          "email": "malak@hotmail.com",
          "positionNumber": 1,
          "isForTrade": true,
          "createdAt": "2025-12-09T20:32:24.568Z",
          "updatedAt": "2025-12-09T20:40:26.572Z",
          "v": 0
        }
      ]
    }
  }
}
```

13. Update Position API

The screenshot shows a Postman interface with a PUT request to the endpoint `http://localhost:5000/api/positions/69388758183e9a9df6f3a6b7`. The request is successful, returning a 200 OK status. The request body is a JSON object with the position number set to 5. The response body is a JSON object with a success message and the updated position details.

```
1 {
  "message": "Position updated",
  "position": {
    "_id": "69388758183e9a9df6f3a6b7",
    "gam3yaId": "69387c55183e9a9df6f3a6ab",
    "userId": "69386d1a183e9a9df6f3a6a2",
    "positionNumber": 5,
    "isForTrade": true,
    "createdAt": "2025-12-09T20:32:24.568Z",
    "updatedAt": "2025-12-09T20:50:16.787Z",
    "v": 0
  }
}
```

14. Delete Positions API

The screenshot shows the Postman interface with a DELETE request to the endpoint `http://localhost:5000/api/positions/69388758183e9a9df6f3a6b7`. The request is configured with the following settings:

- Method: DELETE
- URL: `http://localhost:5000/api/positions/69388758183e9a9df6f3a6b7`
- Body: None
- Status: 200 OK
- Time: 233 ms
- Size: 297 B

The response body is displayed in JSON format:

```
1 { "message": "Position deleted" }
```

15. Buy Position API

The screenshot shows the Postman interface with a POST request to the endpoint `http://localhost:5000/api/positions/69388f894328f9a2bf3636da/buy`. The request is configured with the following settings:

- Method: POST
- URL: `http://localhost:5000/api/positions/69388f894328f9a2bf3636da/buy`
- Body: Raw (JSON)
- Status: 200 OK
- Time: 462 ms
- Size: 1000 B

The request body is:

```
1 {  
2   "newUserId": "69386d1a183e9a9df6f3a6a2"  
3 }
```

The response body is displayed in JSON format:

```
1 { "message": "Position purchased", "position": { "_id": "69388f894328f9a2bf3636da", "gam3yaId":  
  { "_id": "69387c55183e9a9df6f3a6ab", "name": "JanuaryPoolUpdated", "userId":  
    { "_id": "69386d1a183e9a9df6f3a6a2", "username": "malak", "email": "malak@hotmail.com" },  
    "positionNumber": 1, "isForTrade": false, "createdAt": "2025-12-09T21:07:21.280Z",  
    "updatedAt": "2025-12-09T21:45:31.041Z", "_v": 0 }, "transactionDetails": { "position":  
      { "positionId": "69388f894328f9a2bf3636da", "positionNumber": 1, "gam3yaId": "69387c55183e9a9df6f3a6ab",  
        "gam3yaName": "JanuaryPoolUpdated", "previousOwner": { "userId": "69386d1a183e9a9df6f3a6a2",  
          "username": "malak", "email": "malak@hotmail.com" }, "newOwner": { "userId": "69386d1a183e9a9df6f3a6a2",  
            "username": "malak", "email": "malak@hotmail.com" } } } } }
```


16. Swap Position API

The image shows a Postman interface for testing a REST API. The request is a POST to `http://localhost:5000/api/positions/swap`. The body is raw JSON with the following content:

```
{  "position1Id": "69388f894328f9a2bf3636da",  "position2Id": "693899cfacd06adf9fe7a382"}
```

The response status is 200 OK. The response body is JSON, showing a successful swap operation:

```
{  "message": "Positions swapped successfully",  "swapDetails": {    "position1": {      "positionId": "69388f894328f9a2bf3636da",      "positionNumber": 1,      "gam3yaId": "69387c55183e9a9df6f3a6ab",      "gam3yaName": "JanuaryPoolUpdated",      "previousOwner": {        "userId": "69386d1a183e9a9df6f3a6a2",        "username": "malak",        "email": "malak@hotmail.com"      },      "newOwner": {        "userId": "69389075e3f3901a71816fea",        "username": "leila",        "email": "leila@hotmail.com"      }    },    "position2": {      "positionId": "693899cfacd06adf9fe7a382",      "positionNumber": 2,      "gam3yaId": "69387c55183e9a9df6f3a6ab",      "gam3yaName": "JanuaryPoolUpdated",      "previousOwner": {        "userId": "69389075e3f3901a71816fea",        "username": "leila",        "email": "leila@hotmail.com"      },      "newOwner": {        "userId": "69386d1a183e9a9df6f3a6a2",        "username": "malak",        "email": "malak@hotmail.com"      }    },    "positions": [      {        "_id": "69388f894328f9a2bf3636da",        "gam3yaId": "69387c55183e9a9df6f3a6ab",        "name": "JanuaryPoolUpdated",        "userId": {          "_id": "69389075e3f3901a71816fea",          "username": "leila",          "email": "leila@hotmail.com"        },        "positionNumber": 1,        "isForTrade": false,        "createdAt": "2025-12-09T21:07:21.280Z",        "updatedAt": "2025-12-09T22:03:41.355Z",        "_v": 0      },      {        "_id": "693899cfacd06adf9fe7a382",        "gam3yaId": {          "_id": "69387c55183e9a9df6f3a6ab",          "name": "JanuaryPoolUpdated"        },        "userId": {          "_id": "69386d1a183e9a9df6f3a6a2",          "username": "malak",          "email": "malak@hotmail.com"        },        "positionNumber": 2,        "isForTrade": false,        "createdAt": "2025-12-09T21:51:11.125Z",        "updatedAt": "2025-12-09T22:03:41.414Z",        "_v": 0      }    ]  }}
```