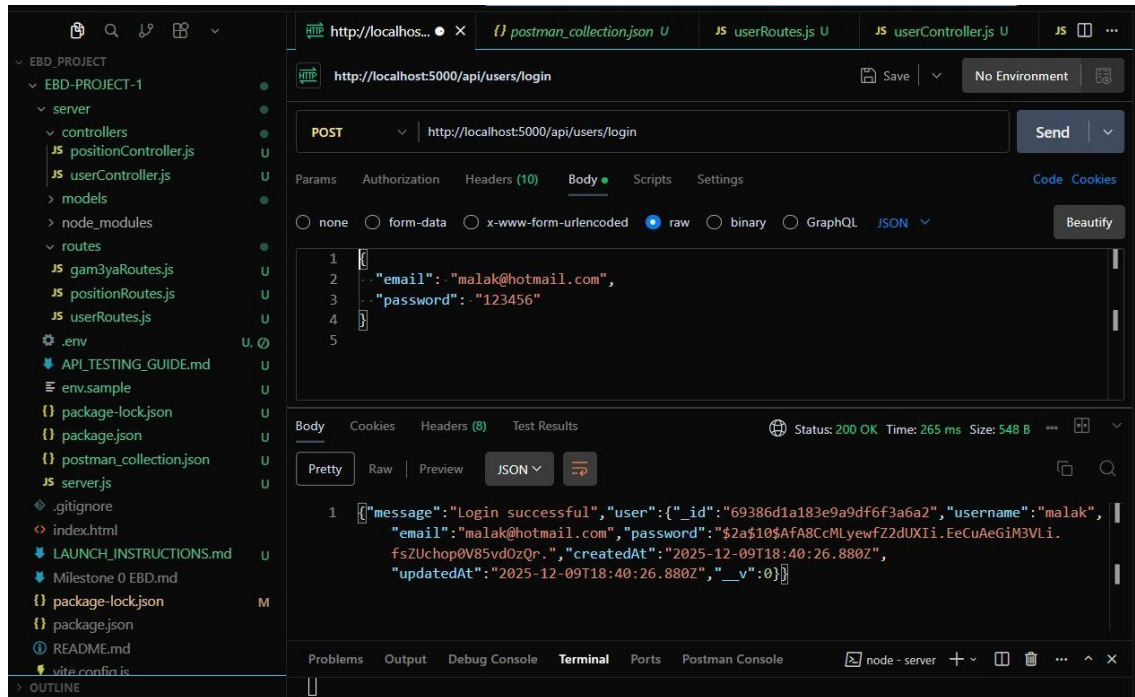
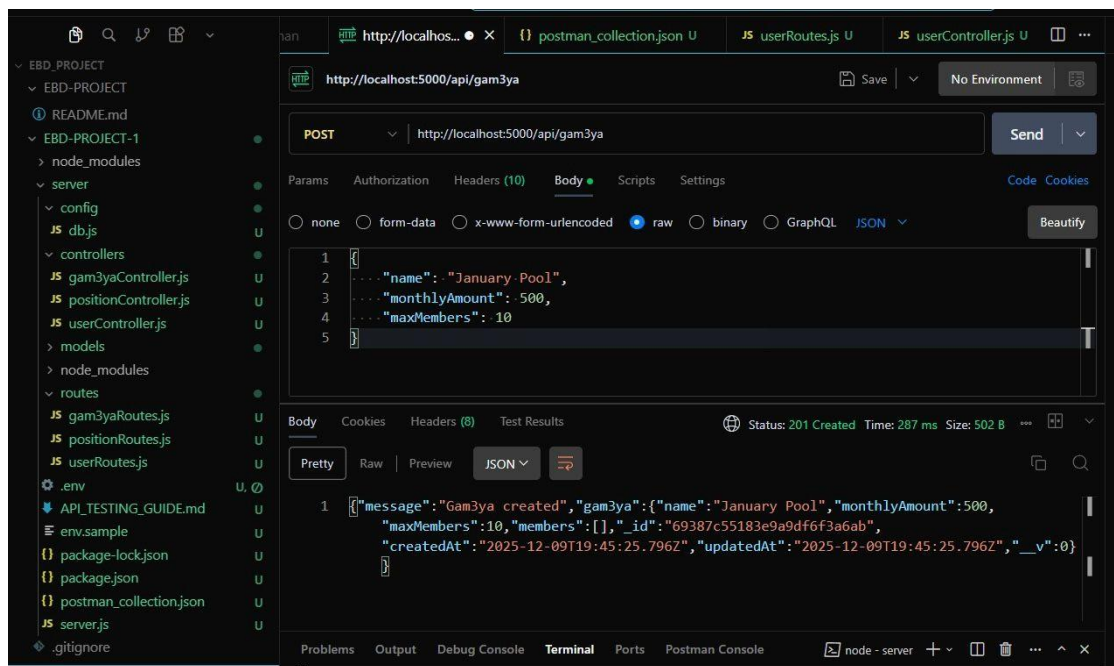


API testing using postman screenshots:

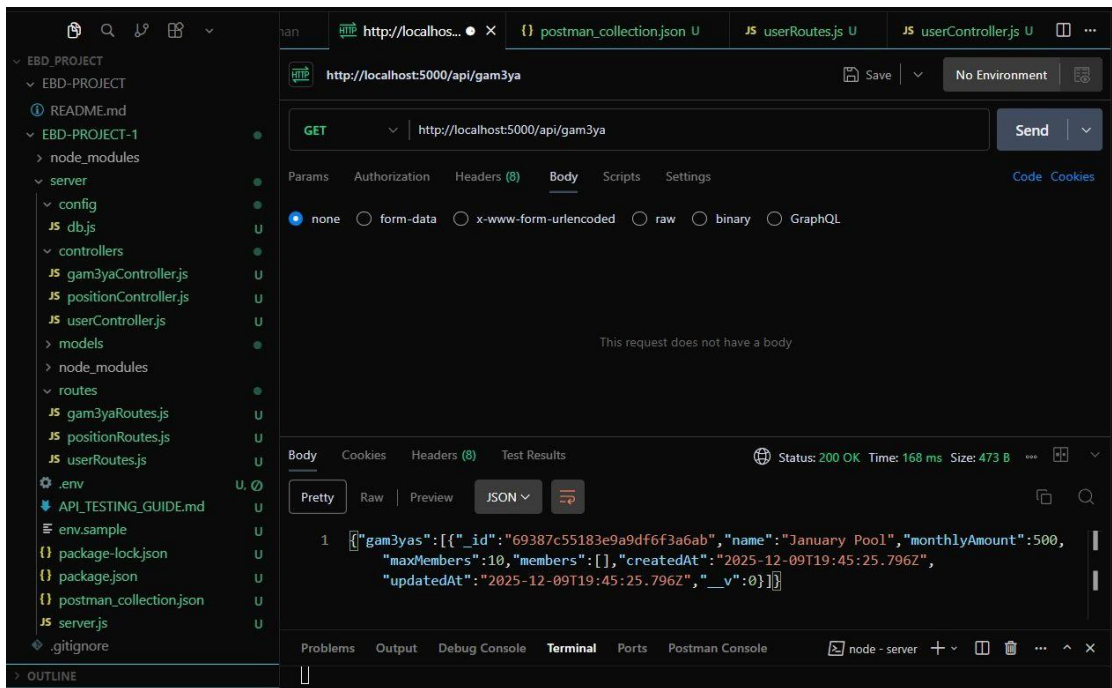
1. Login API



2. Create Gam3eya API



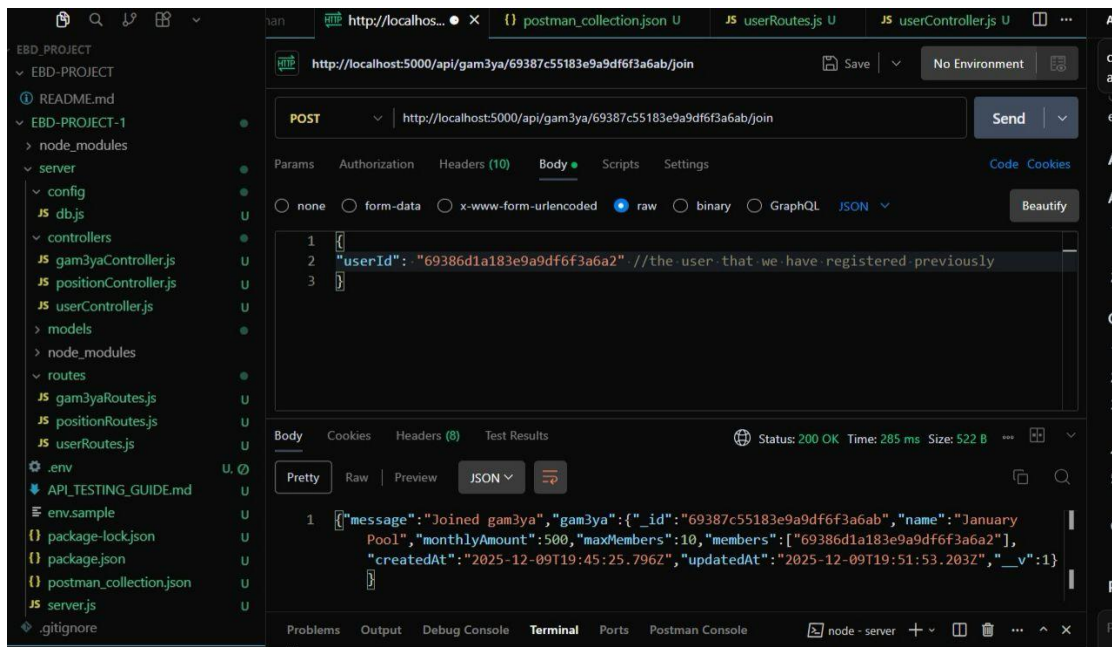
3. Get All Gam3eyas API



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, and the response body is displayed in the JSON tab.

```
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, and the response body is displayed in the JSON tab.

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

5. Get Gam3eya By ID API

The screenshot shows the Postman interface with a GET request to `http://localhost:5000/api/gam3ya/69387c55183e9a9df6f3a6ab`. The request is successful, returning a 200 OK status. The response body is a JSON object representing a Gam3eya entity.

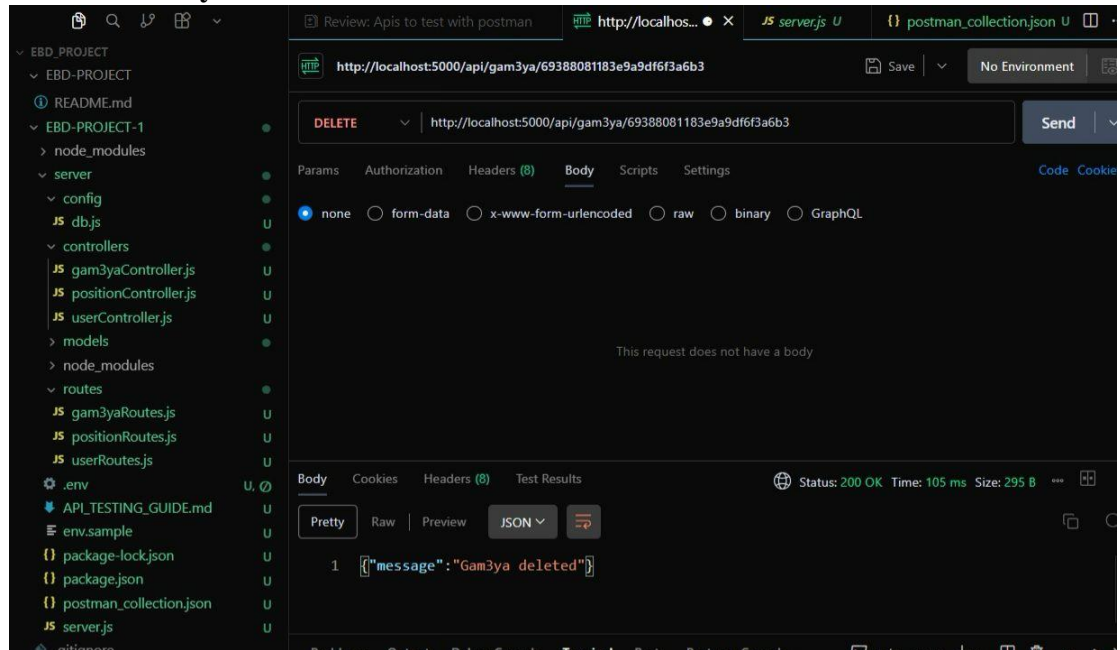
```
{
  "gam3ya": {
    "_id": "69387c55183e9a9df6f3a6ab",
    "name": "January Pool",
    "monthlyAmount": 500,
    "maxMembers": 10,
    "members": [
      {
        "_id": "69386d1a183e9a9df6f3a6a2",
        "username": "malak",
        "email": "malak@hotmail.com",
        "createdAt": "2025-12-09T19:45:25.796Z",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

6. Update Gam3eya API

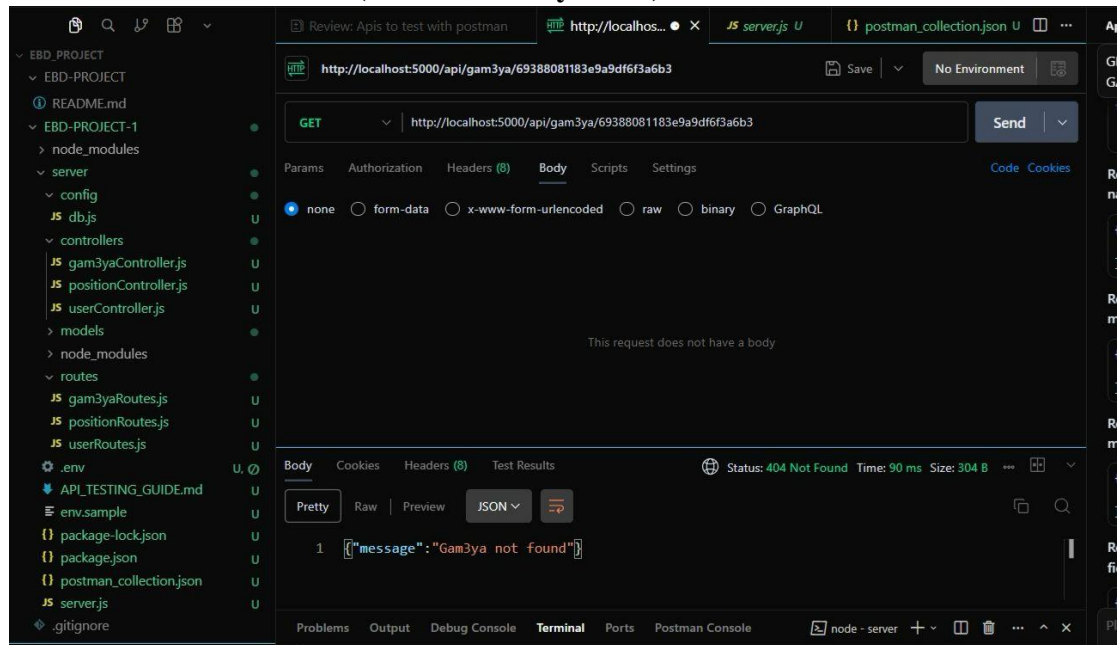
The screenshot shows the Postman interface with a PUT request to `http://localhost:5000/api/gam3ya/69387c55183e9a9df6f3a6ab`. The request is successful, returning a 200 OK status. The response body is a JSON object containing a message and the updated Gam3eya entity.

```
{
  "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
      }
    ]
  }
}
```

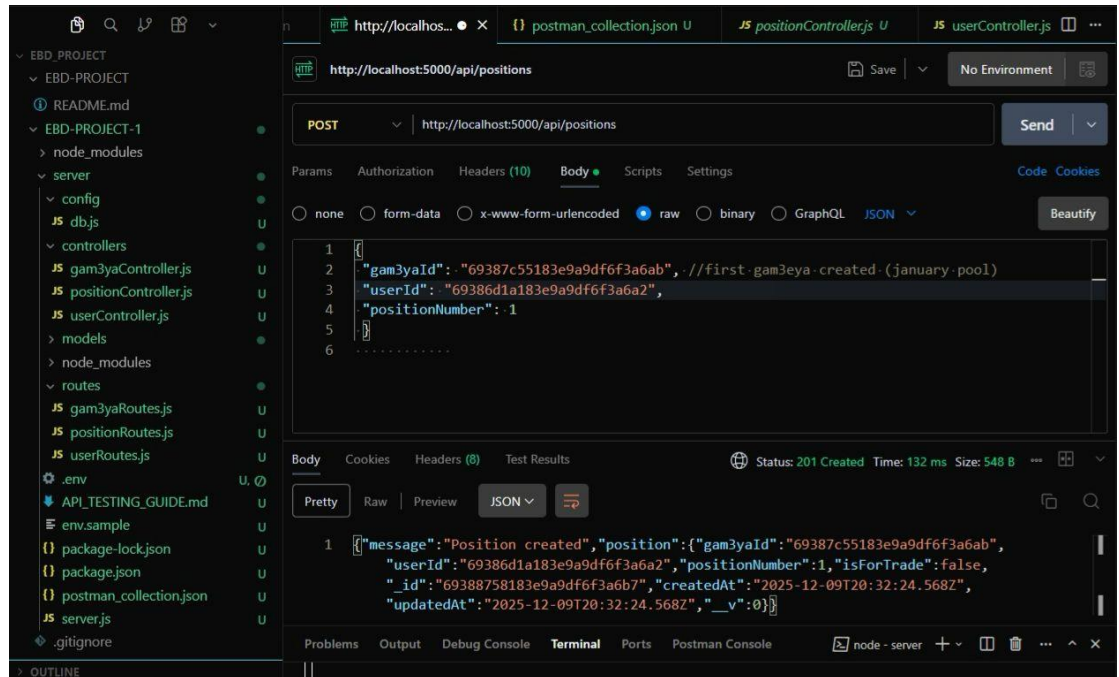
7. Delete Gam3eya API



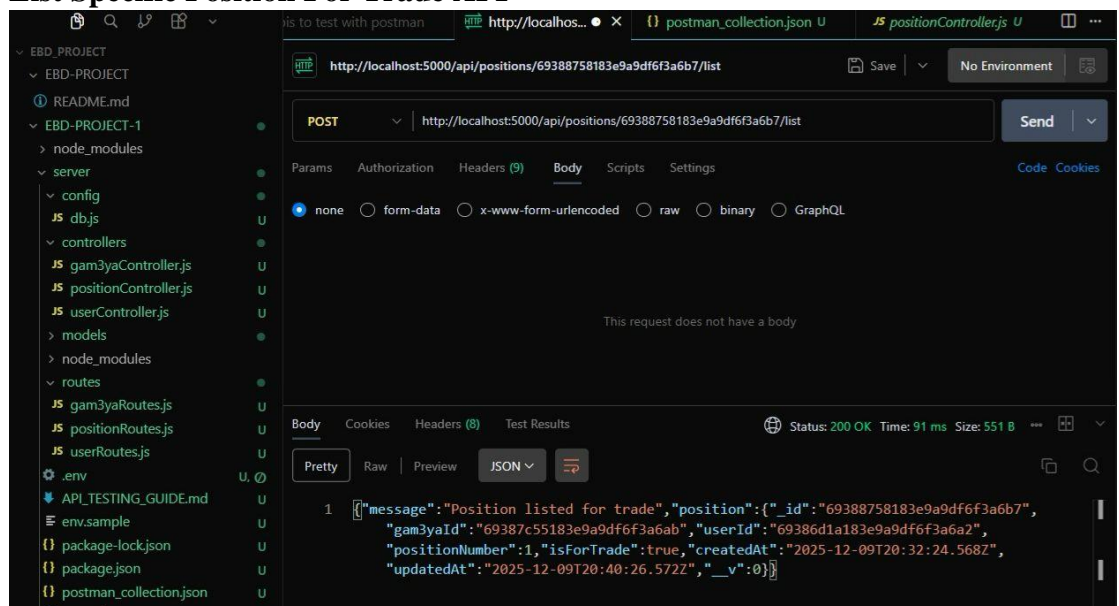
Confirmation of deletion (Delete Gam3eya API)



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": ["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": ["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": {
      "_id": "69387c55183e9a9df6f3a6ab",
      "username": "malak",
      "email": "malak@hotmail.com",
      "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 project named 'EBD_PROJECT'. The left sidebar lists the project structure, including files like 'README.md', 'EBD-PROJECT-1', 'node_modules', 'server', 'config', 'db.js', 'controllers', 'gam3yaController.js', 'positionController.js', 'userController.js', 'models', 'node_modules', 'routes', 'gam3yaRoutes.js', 'positionRoutes.js', 'userRoutes.js', '.env', 'API_TESTING_GUIDE.md', 'env.sample', 'package-lock.json', 'package.json', 'postman_collection.json', 'server.js', and '.gitignore'. The main panel displays a DELETE request to the URL 'http://localhost:5000/api/positions/69388758183e9a9df6f3a6b7'. The request is configured with the method 'DELETE' and the body is empty. The status bar at the bottom indicates a successful response: 'Status: 200 OK Time: 233 ms Size: 297 B'. The response body is shown in the 'Body' tab, displaying a JSON object: {'message': 'Position deleted'}

15. Buy Position API

The screenshot shows the Postman interface with a project named 'EBD_PROJECT'. The left sidebar lists the project structure, including files like 'README.md', 'EBD-PROJECT-1', 'node_modules', 'server', 'config', 'db.js', 'controllers', 'gam3yaController.js', 'positionController.js', 'userController.js', 'models', 'node_modules', 'routes', 'gam3yaRoutes.js', 'positionRoutes.js', 'userRoutes.js', '.env', 'API_TESTING_GUIDE.md', 'env.sample', 'package-lock.json', 'package.json', 'postman_collection.json', 'server.js', and '.gitignore'. The main panel displays a POST request to the URL 'http://localhost:5000/api/positions/69388f894328f9a2bf3636da/buy'. The request is configured with the method 'POST' and the body is set to 'raw'. The status bar at the bottom indicates a successful response: 'Status: 200 OK Time: 462 ms Size: 1000 B'. The response body is shown in the 'Body' tab, displaying a JSON object: {'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:

```
1 {
2   "position1Id": "69388f894328f9a2bf3636da",
3   "position2Id": "693899cfacd06adf9fe7a382"
4 }
```

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

```
1 {
2   "message": "Positions swapped successfully",
3   "swapDetails": {
4     "position1": {
5       "positionId": "69388f894328f9a2bf3636da",
6       "positionNumber": 1,
7       "gam3yaId": "69387c55183e9a9df6f3a6ab",
8       "gam3yaName": "JanuaryPoolUpdated",
9       "previousOwner": {
10        "userId": "69386d1a183e9a9df6f3a6a2",
11        "username": "malak",
12        "email": "malak@hotmail.com"
13      },
14       "newOwner": {
15        "userId": "69389075e3f3901a71816fea",
16        "username": "leila",
17        "email": "leila@hotmail.com"
18      }
19     },
20     "position2": {
21       "positionId": "693899cfacd06adf9fe7a382",
22       "positionNumber": 2,
23       "gam3yaId": "69387c55183e9a9df6f3a6ab",
24       "gam3yaName": "JanuaryPoolUpdated",
25       "previousOwner": {
26        "userId": "69389075e3f3901a71816fea",
27        "username": "leila",
28        "email": "leila@hotmail.com"
29      },
30       "newOwner": {
31        "userId": "69386d1a183e9a9df6f3a6a2",
32        "username": "malak",
33        "email": "malak@hotmail.com"
34      }
35     },
36     "positions": [
37       {
38         "_id": "69388f894328f9a2bf3636da",
39         "gam3yaId": "69387c55183e9a9df6f3a6ab",
40         "name": "JanuaryPoolUpdated",
41         "userId": {
42           "_id": "69389075e3f3901a71816fea",
43           "username": "leila",
44           "email": "leila@hotmail.com"
45         },
46         "positionNumber": 1,
47         "isForTrade": false,
48         "createdAt": "2025-12-09T21:07:21.280Z",
49         "updatedAt": "2025-12-09T22:03:41.355Z",
50         "_v": 0
51       },
52       {
53         "_id": "693899cfacd06adf9fe7a382",
54         "gam3yaId": {
55           "_id": "69387c55183e9a9df6f3a6ab",
56           "name": "JanuaryPoolUpdated"
57         },
58         "userId": {
59           "_id": "69386d1a183e9a9df6f3a6a2",
60           "username": "malak",
61           "email": "malak@hotmail.com"
62         },
63         "positionNumber": 2,
64         "isForTrade": false,
65         "createdAt": "2025-12-09T21:51:11.125Z",
66         "updatedAt": "2025-12-09T22:03:41.414Z",
67         "_v": 0
68       }
69     ]
70   }
71 }
```