HOTELS

In Business since 1995

Service Layer

Majed Hassan

November 05, 2023

Overview

Route Layer:

In the route layer of the backend of thisr hotel website, this will create particular paths (URL endpoints) that the frontend can use to communicate with the server. These routes are accountable for processing HTTP requests and directing them to the relevant controllers in the system. This can provide access to certain information and capabilities related to hotels, bookings, and user administration by defining routes such as /api/user, /api/bookings, and /api/category, for example. These routes can be used to provide access to specific information.

Controller Layer:

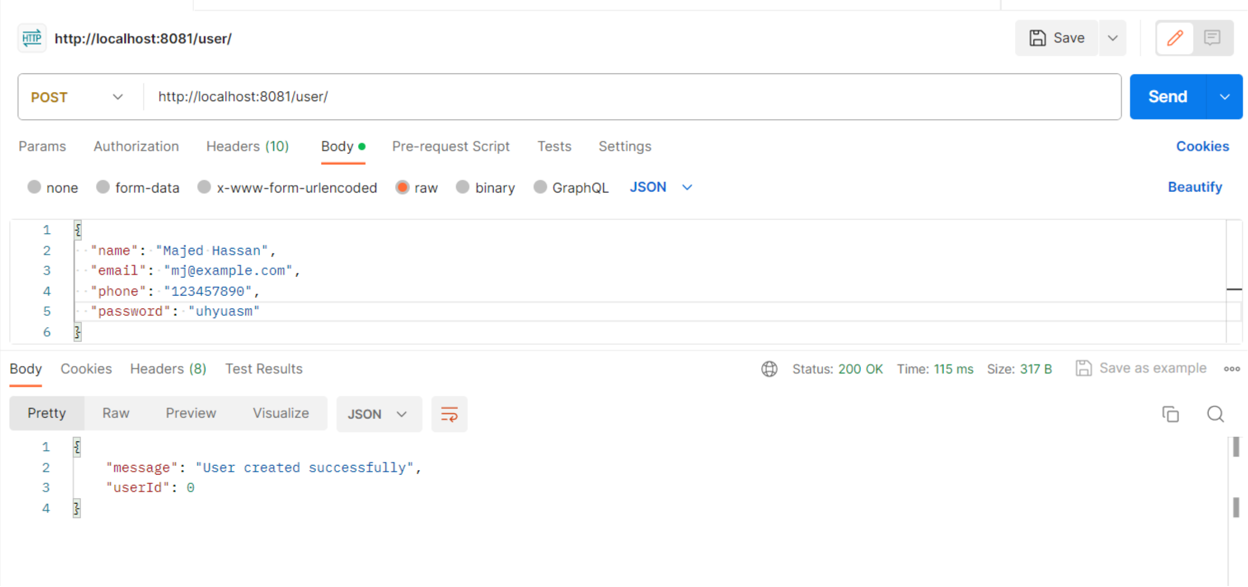
In the controller layer, this will implement the logic that will be used to handle incoming HTTP requests and return appropriate responses to the frontend of the application. Each route that establishes in the route layer will eventually lead to a particular controller, also known as an endpoint. This may, for instance, have a hotelsController that handles requests pertaining to hotels, a bookingsController that manages bookings, and a usersController that handles actions pertaining to users. These controllers take in incoming requests, perform necessary data processing, contact the appropriate service methods, and then relay the results of their work to the client. The controllers act as a connecting point between the routes and the service layer of the network.

Service Layer:

The service layer of the hotel website backend and the business logic of the website. Each controller is responsible for delegating its activities to the appropriate service methods. In order to obtain, alter, or otherwise work with the data, the service methods connect with the database, which serves as the repository layer. For instance, this may have methods within the hotelsService that allow this to retrieve hotel information, add new hotels, update current hotels, or delete hotels. In a similar way, the bookingsService would be in charge of handling booking-related tasks such as checking availability, making reservations, and canceling existing bookings. Using these service methods, this can be certain that the fundamental functionality of this application is kept entirely distinct from the routing and controller logic.

Method: Post

Purpose: New users or guests can be added to the hotel's system via the API's "create user" method. This key action enables account creation, which in turn enables reservation making, individualized service delivery, and data management for individual users.



http://localhost:8081/user/

POST http://localhost:8081/user/

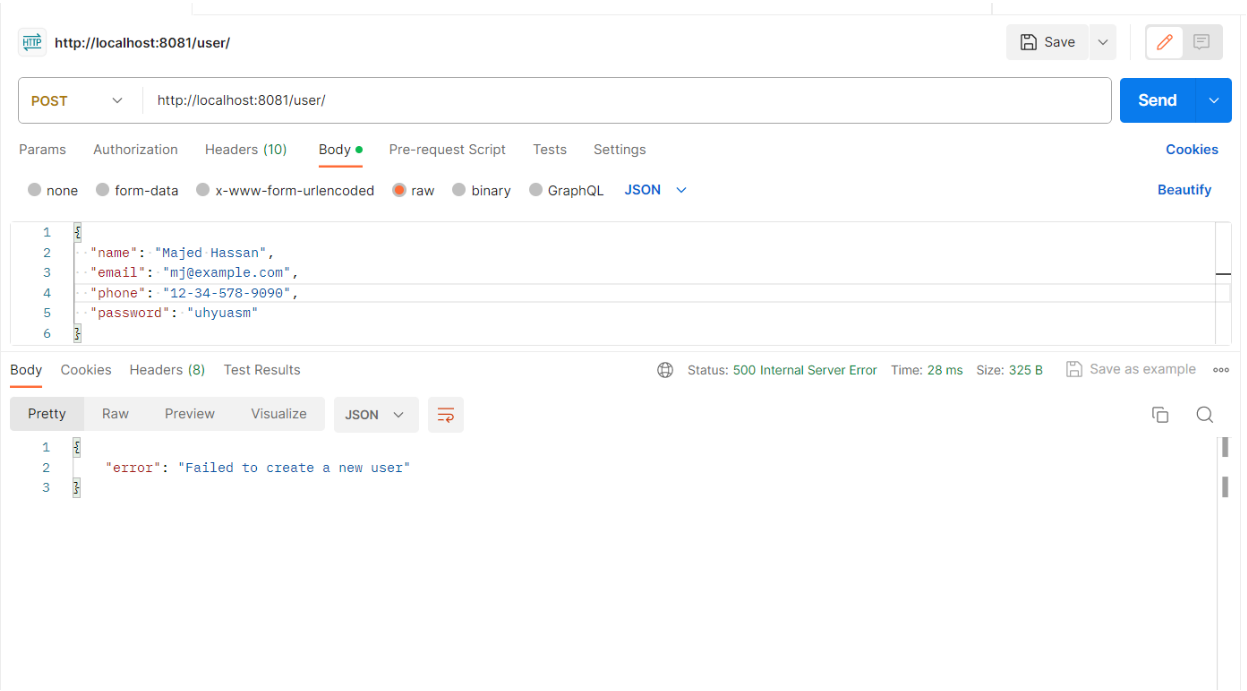
```
{
  "name": "Majed Hassan",
  "email": "mj@example.com",
  "phone": "123457890",
  "password": "uhyuasm"
}
```

Status: 200 OK Time: 115 ms Size: 317 B

JSON

```
{
    "message": "User created successfully",
    "userId": 0
}
```

Error Handling



http://localhost:8081/user/

POST http://localhost:8081/user/ Send

```
{
  "name": "Majed Hassan",
  "email": "mj@example.com",
  "phone": "12-34-578-9090",
  "password": "uhyuasm"
}
```

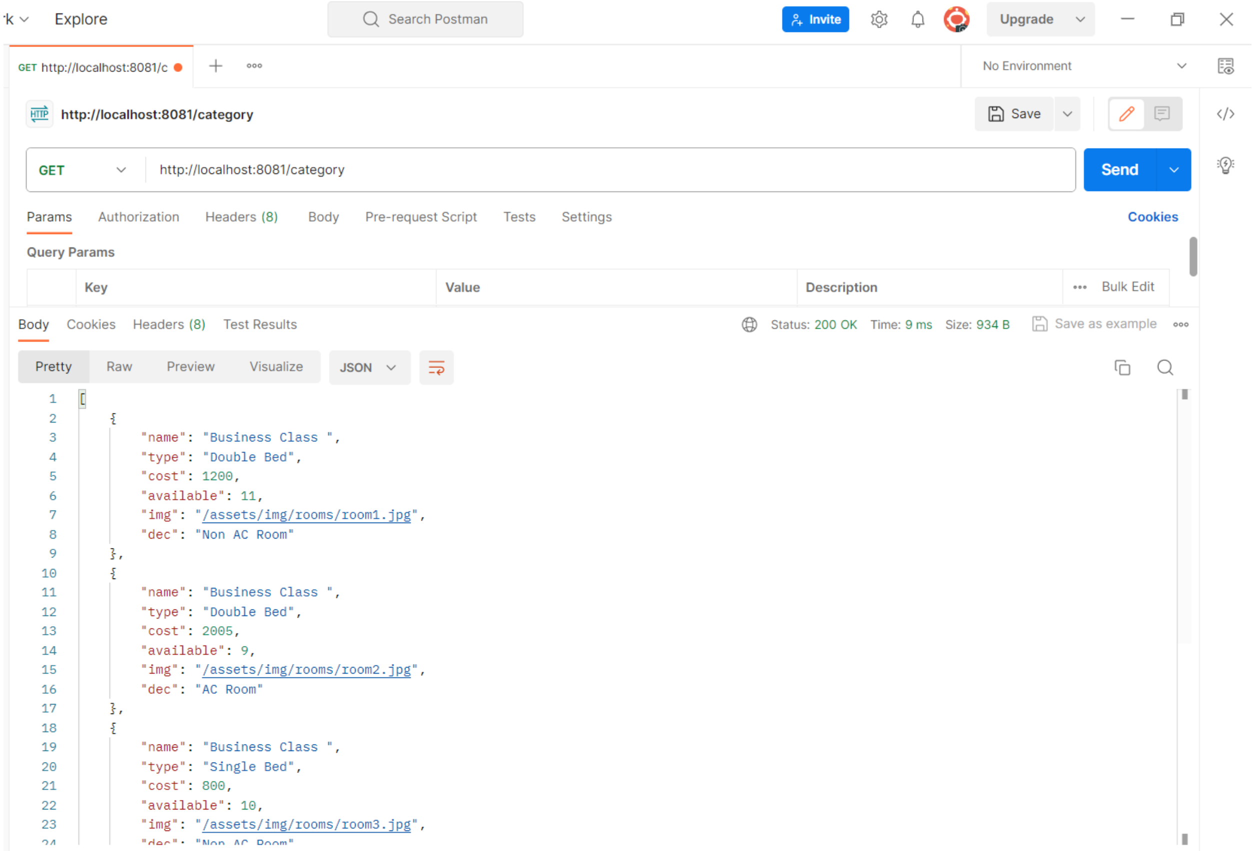
Status: 500 Internal Server Error Time: 28 ms Size: 325 B

JSON

```
{
    "error": "Failed to create a new user"
}
```

Method: Get

Purpose: In an application programming interface (API) designed for a hotel development project, the "get" method has the responsibility of getting information about the various types of rooms, services, and facilities that are provided by the hotel.



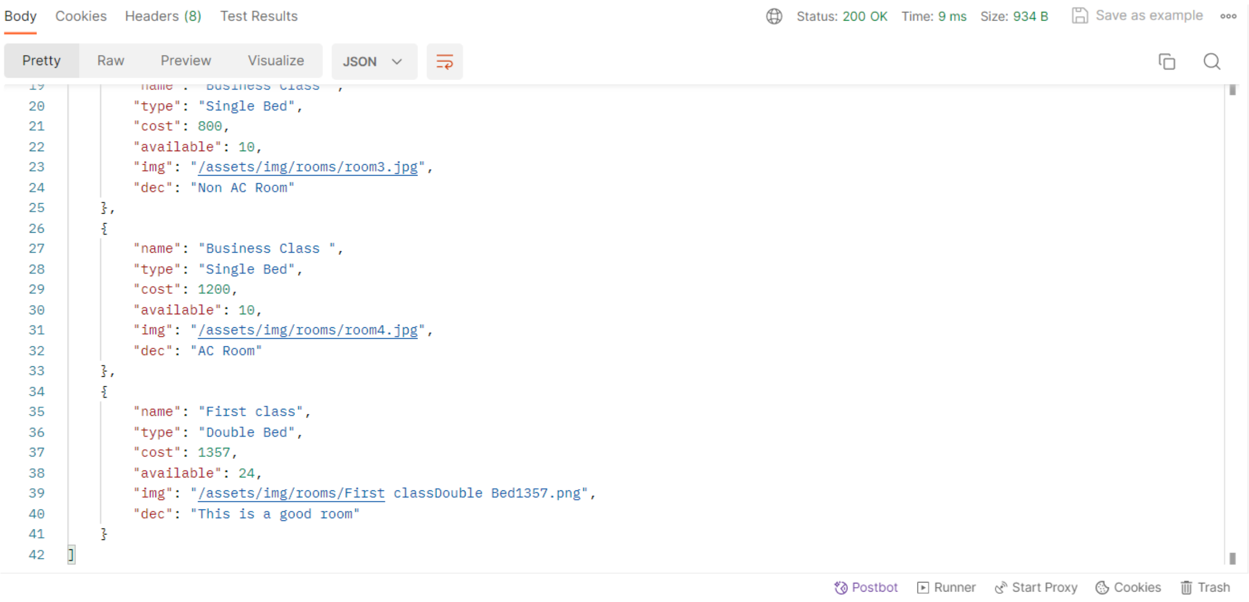
GET http://localhost:8081/c

http://localhost:8081/category

GET http://localhost:8081/category

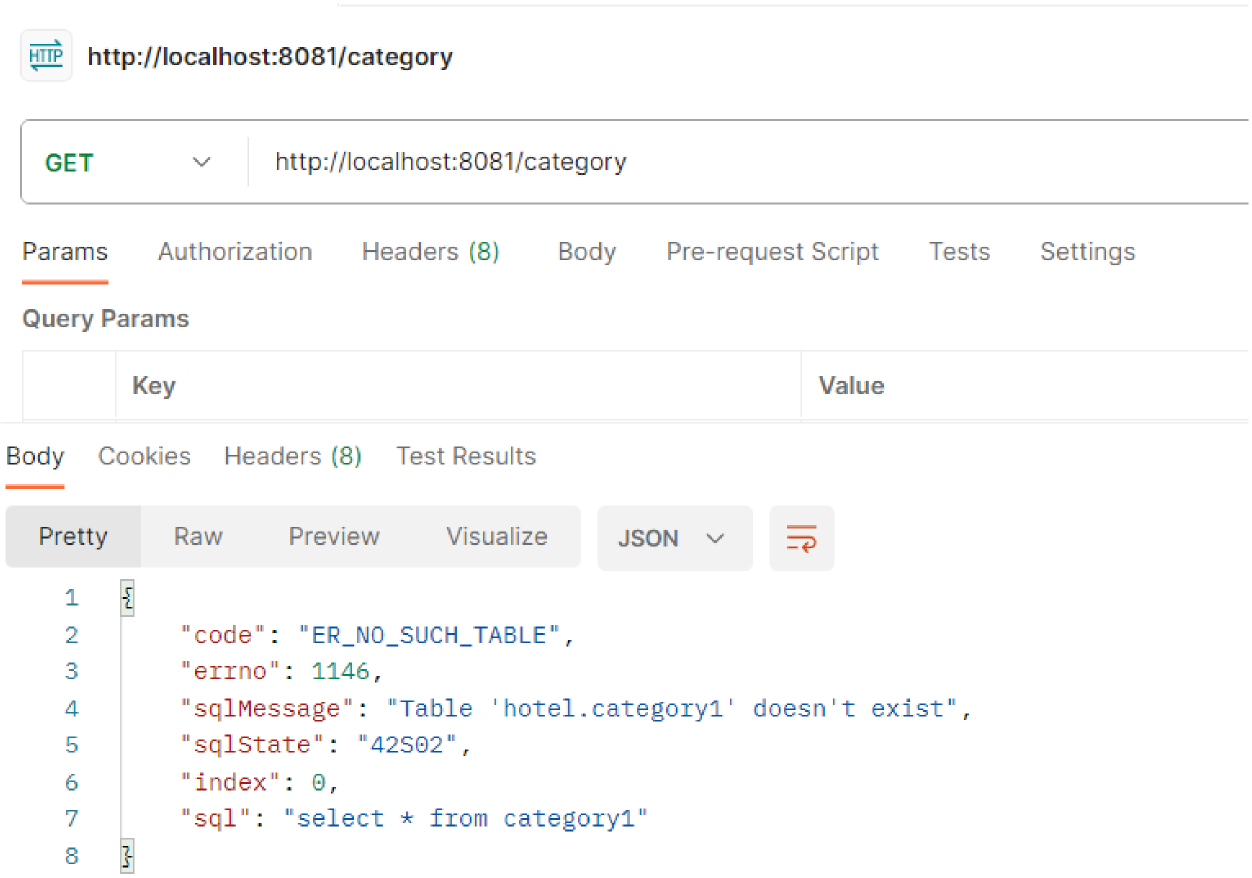
Status: 200 OK Time: 9 ms Size: 934 B

```
[
    {
        "name": "Business Class ",
        "type": "Double Bed",
        "cost": 1200,
        "available": 11,
        "img": "/assets/img/rooms/room1.jpg",
        "dec": "Non AC Room"
    },
    {
        "name": "Business Class ",
        "type": "Double Bed",
        "cost": 2005,
        "available": 9,
        "img": "/assets/img/rooms/room2.jpg",
        "dec": "AC Room"
    },
    {
        "name": "Business Class ",
        "type": "Single Bed",
        "cost": 800,
        "available": 10,
        "img": "/assets/img/rooms/room3.jpg",
        "dec": "Non AC Room"
```



Status: 200 OK Time: 9 ms Size: 934 B

```
        "name": "Business Class ",
        "type": "Single Bed",
        "cost": 800,
        "available": 10,
        "img": "/assets/img/rooms/room3.jpg",
        "dec": "Non AC Room"
    },
    {
        "name": "Business Class ",
        "type": "Single Bed",
        "cost": 1200,
        "available": 10,
        "img": "/assets/img/rooms/room4.jpg",
        "dec": "AC Room"
    },
    {
        "name": "First class",
        "type": "Double Bed",
        "cost": 1357,
        "available": 24,
        "img": "/assets/img/rooms/First classDouble Bed1357.png",
        "dec": "This is a good room"
    }
]
```

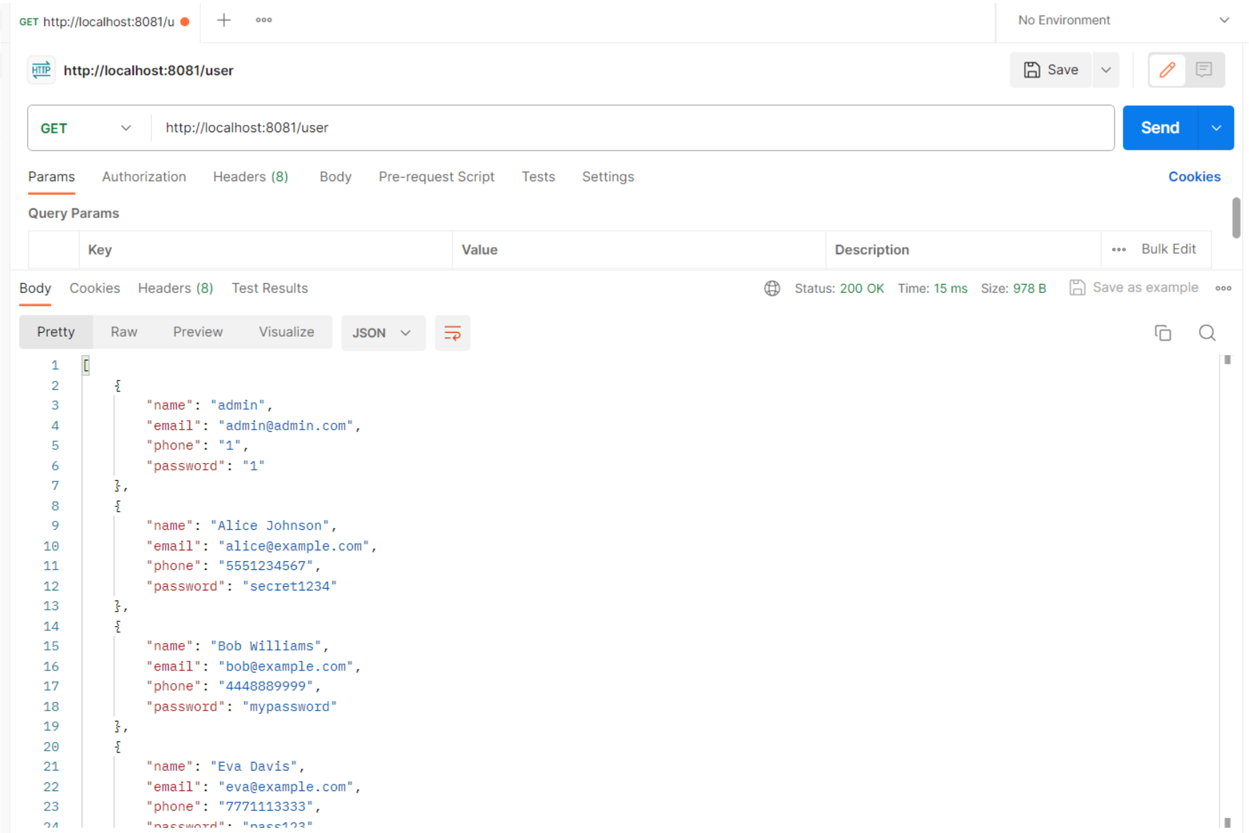


GET http://localhost:8081/category

```
{
    "code": "ER_NO_SUCH_TABLE",
    "errno": 1146,
    "sqlMessage": "Table 'hotel.category1' doesn't exist",
    "sqlState": "42S02",
    "index": 0,
    "sql": "select * from category1"
}
```

Method: Get

Purpose: Admins can access and edit their own profile information, as well as those of their users, with this feature.

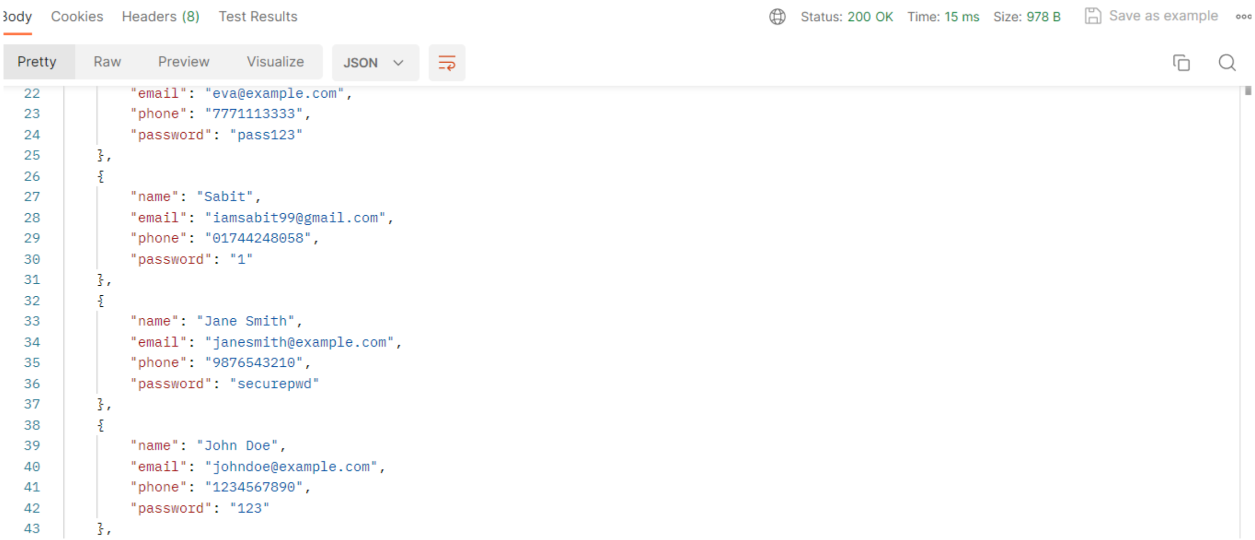


http://localhost:8081/user

GET http://localhost:8081/user

Status: 200 OK Time: 15 ms Size: 978 B

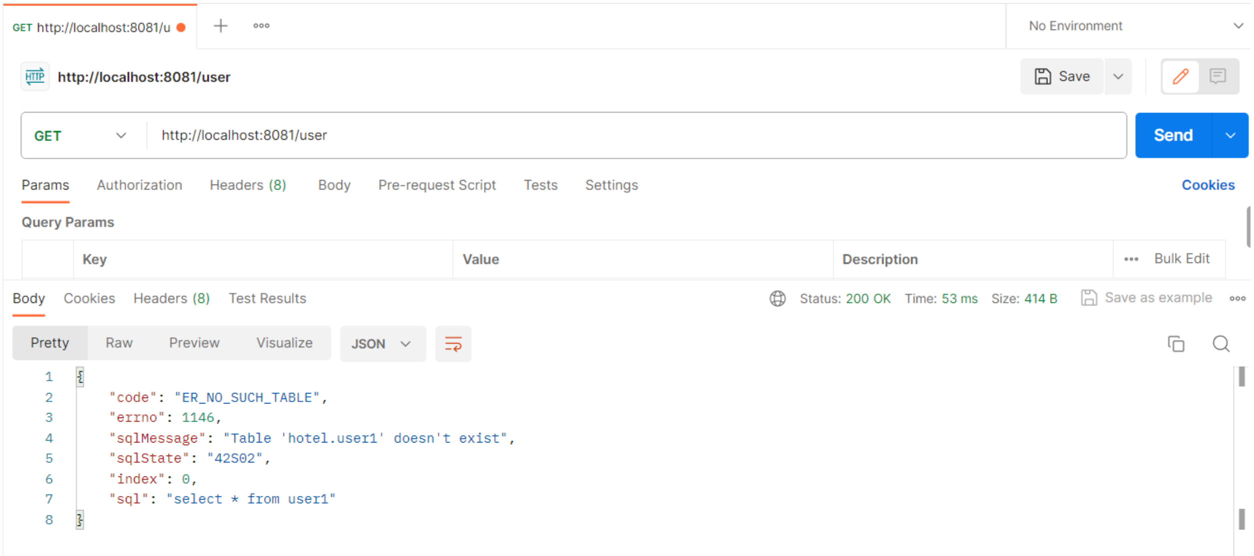
```
[
    {
        "name": "admin",
        "email": "admin@admin.com",
        "phone": "1",
        "password": "1"
    },
    {
        "name": "Alice Johnson",
        "email": "alice@example.com",
        "phone": "5551234567",
        "password": "secret1234"
    },
    {
        "name": "Bob Williams",
        "email": "bob@example.com",
        "phone": "4448889999",
        "password": "mypassword"
    },
    {
        "name": "Eva Davis",
        "email": "eva@example.com",
        "phone": "7771113333",
        "password": "pass123"
```



Status: 200 OK Time: 15 ms Size: 978 B

```
        "email": "eva@example.com",
        "phone": "7771113333",
        "password": "pass123"
    },
    {
        "name": "Sabit",
        "email": "iamsabit99@gmail.com",
        "phone": "01744248058",
        "password": "1"
    },
    {
        "name": "Jane Smith",
        "email": "janesmith@example.com",
        "phone": "9876543210",
        "password": "securepwd"
    },
    {
        "name": "John Doe",
        "email": "johndoe@example.com",
        "phone": "1234567890",
        "password": "123"
    },
```

Error Handling



http://localhost:8081/user

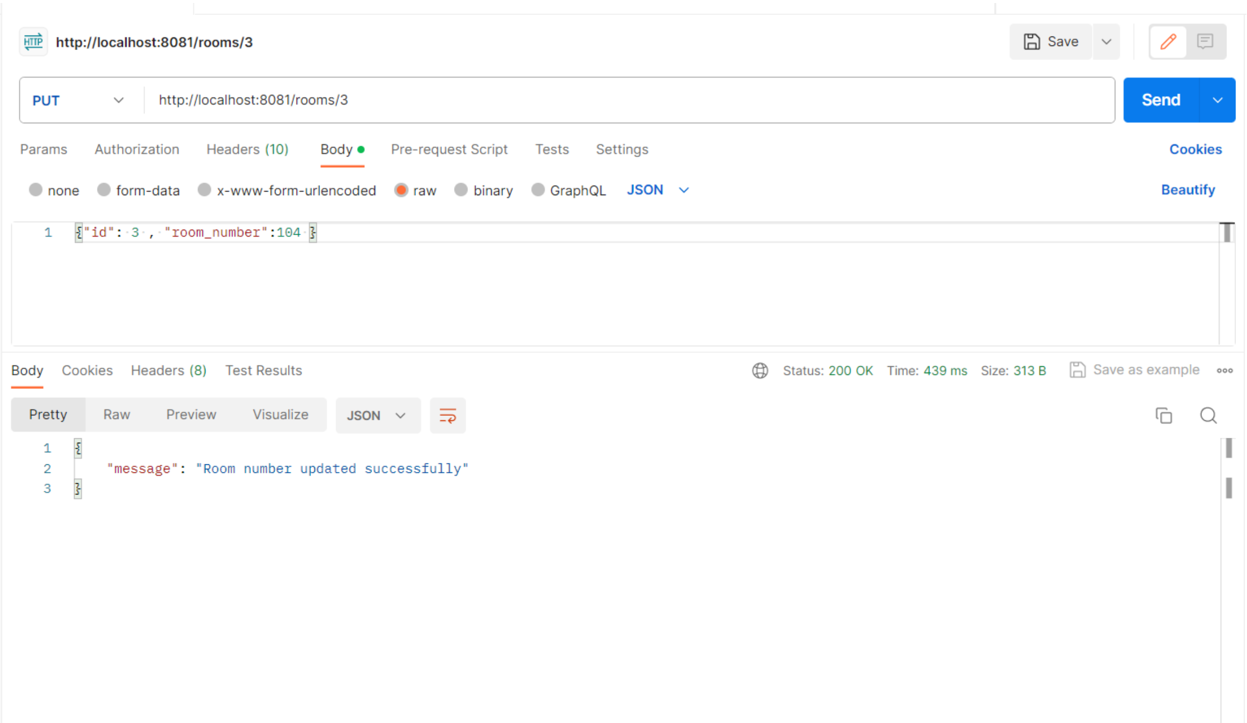
GET http://localhost:8081/user

Status: 200 OK Time: 53 ms Size: 414 B

```
{
    "code": "ER_NO_SUCH_TABLE",
    "errno": 1146,
    "sqlMessage": "Table 'hotel.user1' doesn't exist",
    "sqlState": "42S02",
    "index": 0,
    "sql": "select * from user1"
}
```

Method: Put

Purpose: To make it possible to change and update individual room numbers or details in the hotel's room management system, an API for a hotel project includes a "PUT updating rooms/room_number" function.



PUT http://localhost:8081/rooms/3

Body: raw JSON

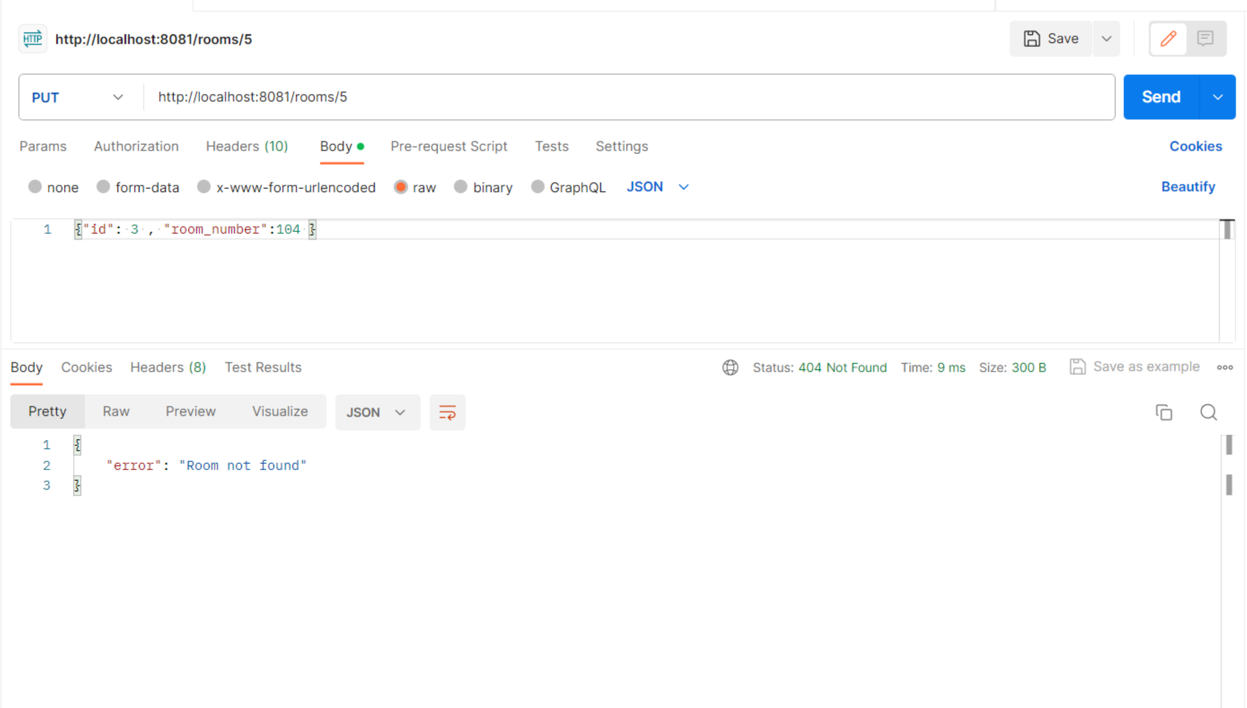
```
{"id": 3 , "room_number":104 }
```

Status: 200 OK Time: 439 ms Size: 313 B

JSON

```
{
    "message": "Room number updated successfully"
}
```

Error Handling



PUT http://localhost:8081/rooms/5

Body: raw JSON

```
{"id": 3 , "room_number":104 }
```

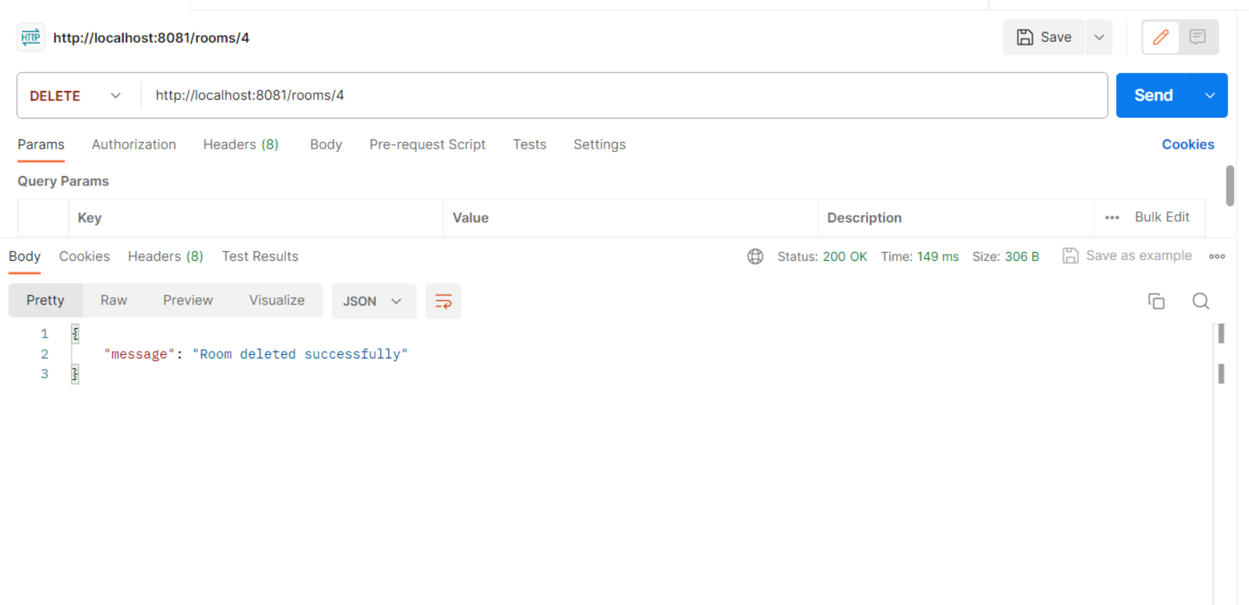
Status: 404 Not Found Time: 9 ms Size: 300 B

JSON

```
{
    "error": "Room not found"
}
```

Method: Delete

Purpose: The system is useful for keeping an updated and precise room count. The hotel's inventory is adjusted to reflect the reduction in available rooms after a room is removed.

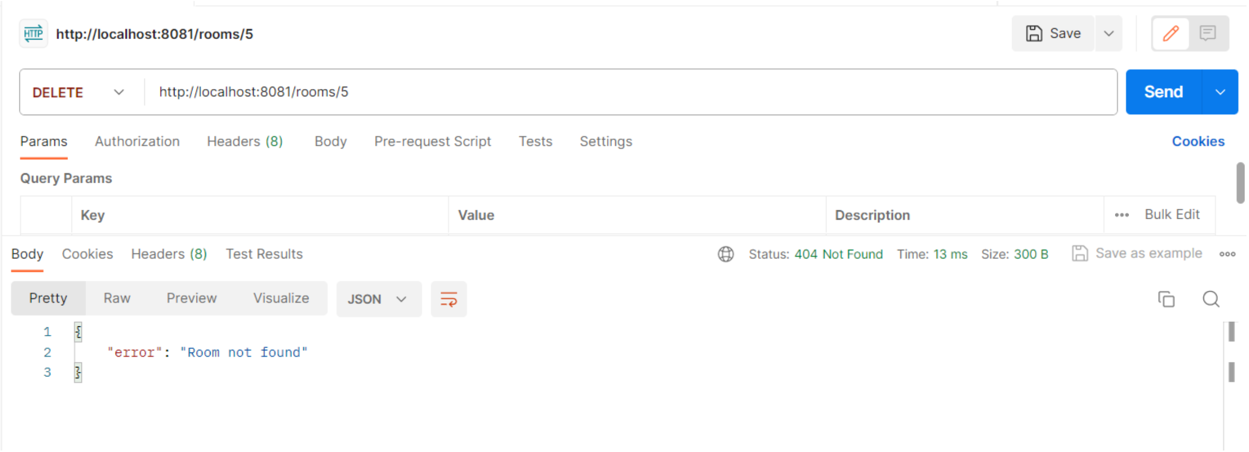


DELETE http://localhost:8081/rooms/4

Status: 200 OK Time: 149 ms Size: 306 B

```
{
    "message": "Room deleted successfully"
}
```

Error Handling

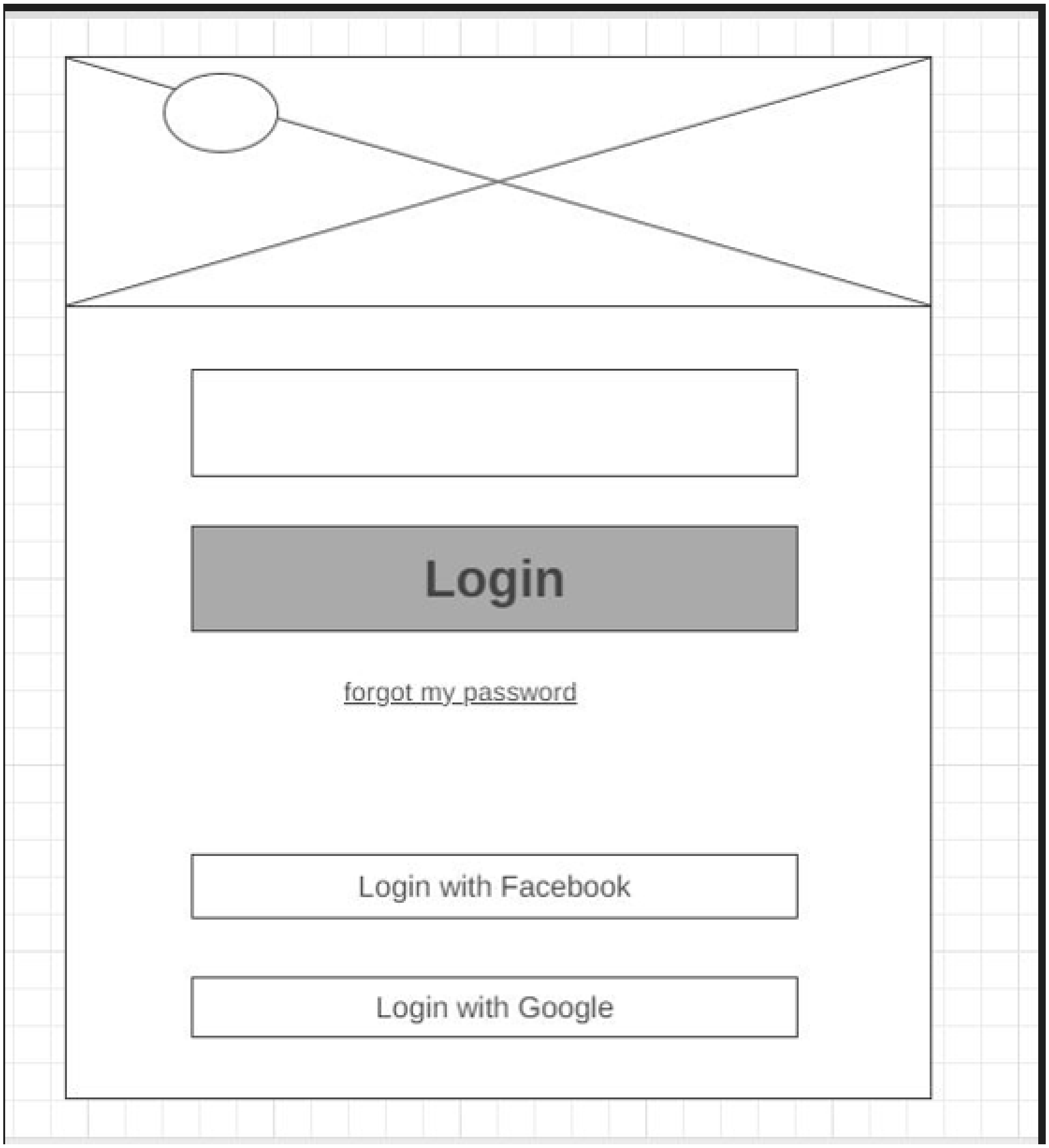


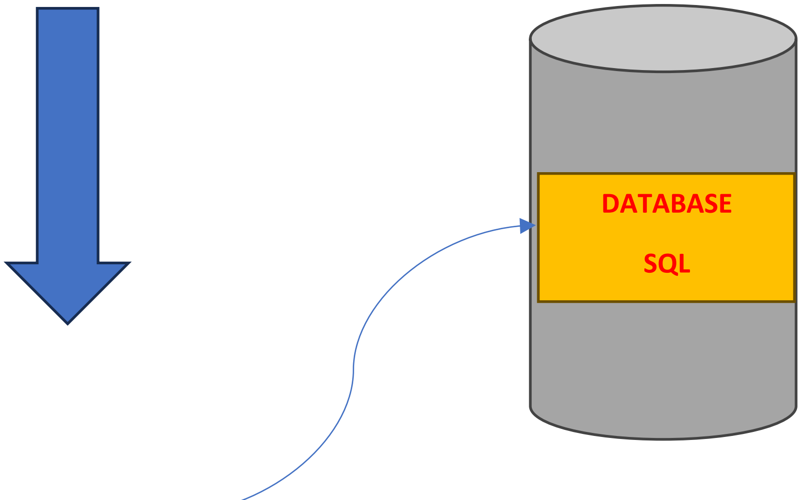
DELETE http://localhost:8081/rooms/5

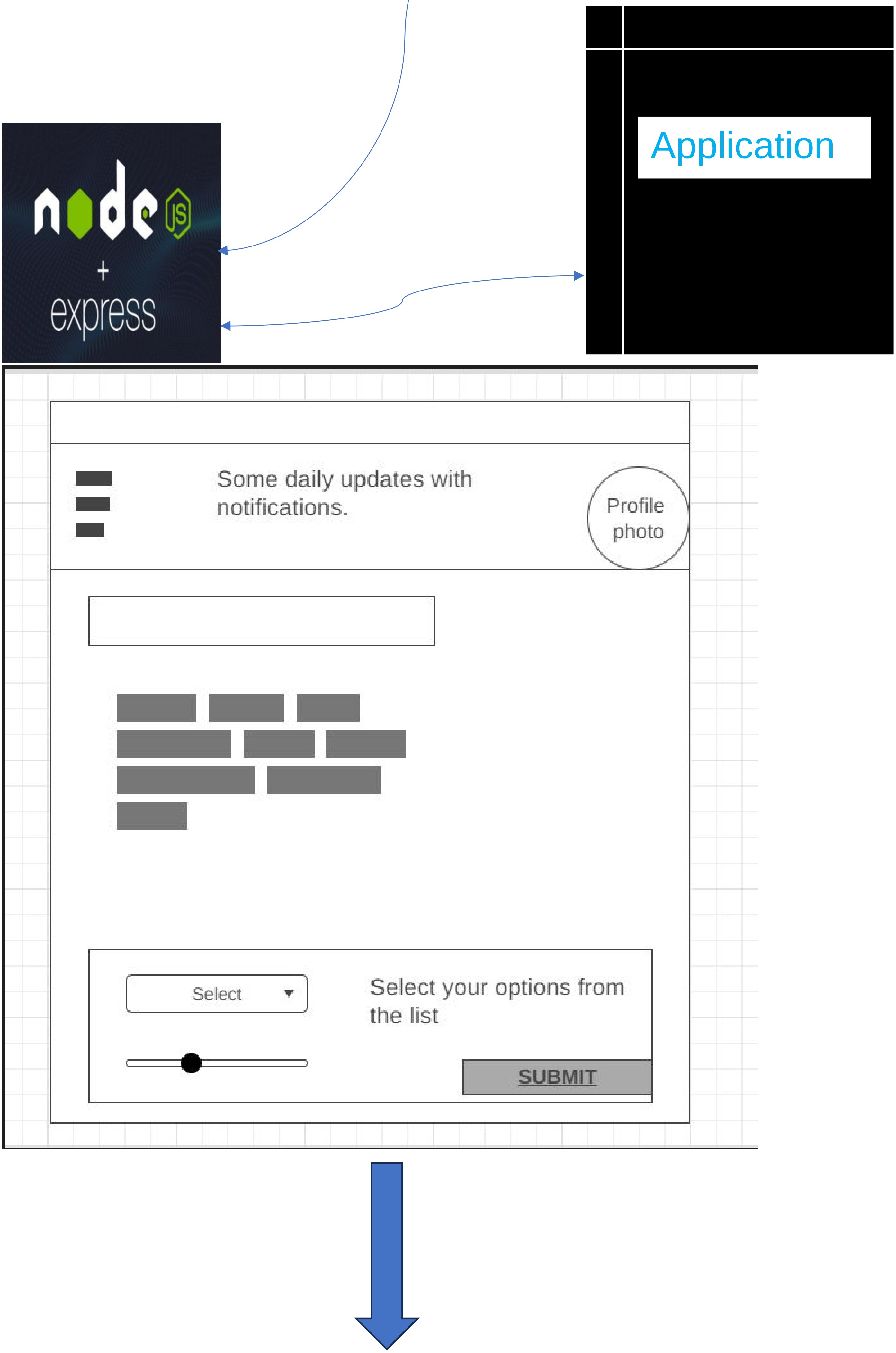
Status: 404 Not Found Time: 13 ms Size: 300 B

```
{
    "error": "Room not found"
}
```

UI Design Planning / Implementation







Application

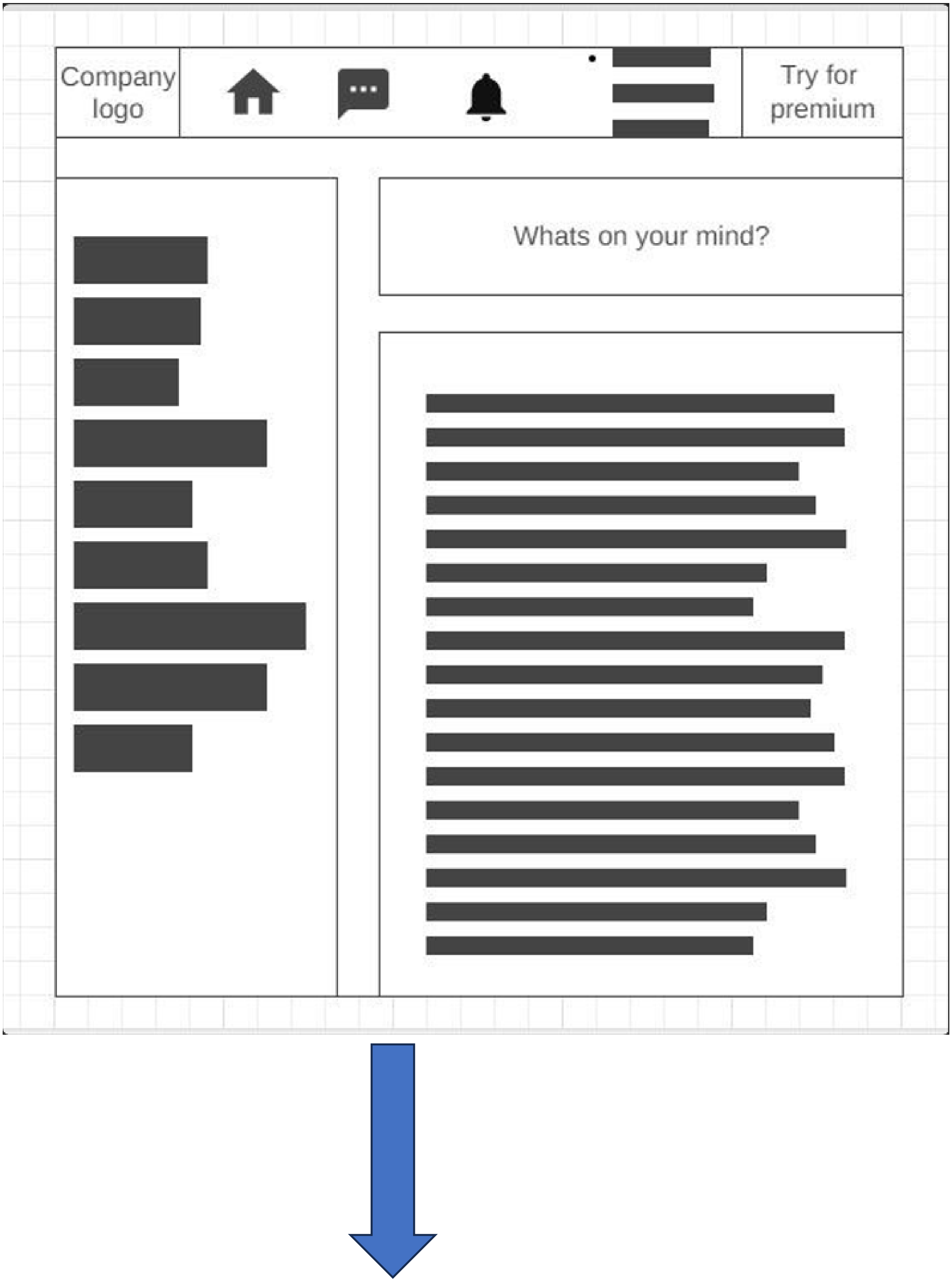
Some daily updates with notifications.

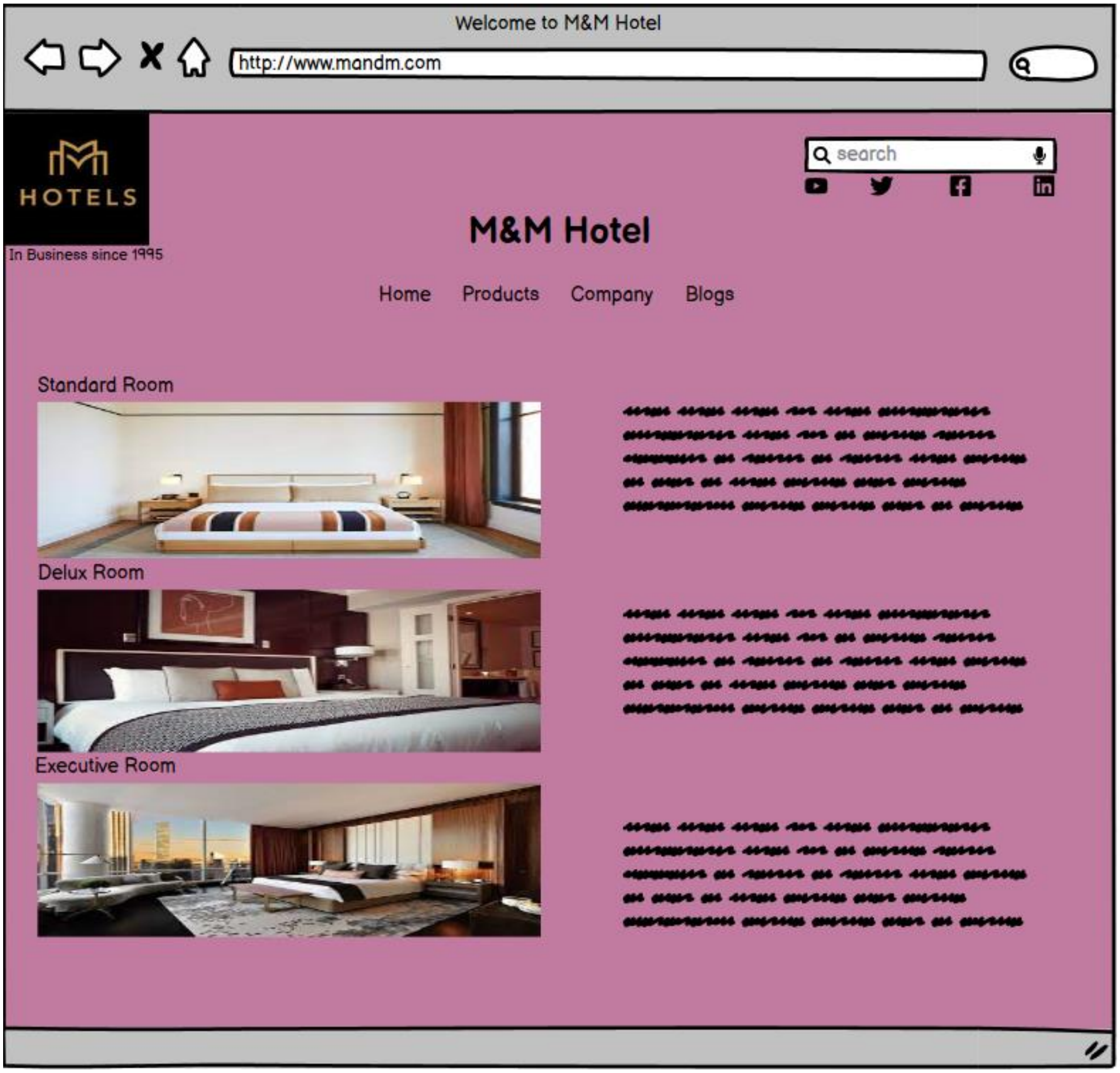
Profile photo

Select ▼

Select your options from the list

SUBMIT


Company logo
Try for premium
Whats on your mind?


Welcome to M&M Hotel
http://www.mandm.com
HOTELS
In Business since 1995
search
M&M Hotel
Home
Products
Company
Blogs
Standard Room
Delux Room
Executive Room