Booking Management System

17-08-2024

PABAU L.L.C

Albion Morina
Software Developer and IT Support
Prishtinë, Kosovë
+38343734849

Overview

This project is a simple booking management system built using Next.js for the frontend and Express.js with MySQL for the backend. The application allows users to create, view, and manage bookings.

Project Goals

1. **Build a functional booking management system** using Next.js and Express.js.
2. **Implement a RESTful API** for creating, viewing, and managing bookings.
3. **Ensure data persistence and integrity** with a MySQL database.
4. **Containerize the application** using Docker for easy deployment and consistent environments.
5. **Provide a user-friendly interface** for interacting with the booking system.

Specifications of the project

Frontend: Developed with Next.js using TypeScript, featuring pages for creating and listing bookings. Styled with Tailwind CSS for a clean and responsive design.

Backend: Built with Express.js, providing API endpoints for fetching and inserting bookings into a MySQL database.

Database: MySQL handles data storage with tables for managing booking details, including service, doctor name, start and end times, and dates.

Docker: The project is containerized using Docker, with services for the frontend, backend, MySQL, and phpMyAdmin for database management.

Error Handling: Implemented basic error handling for API requests to ensure reliability and smooth user experience.

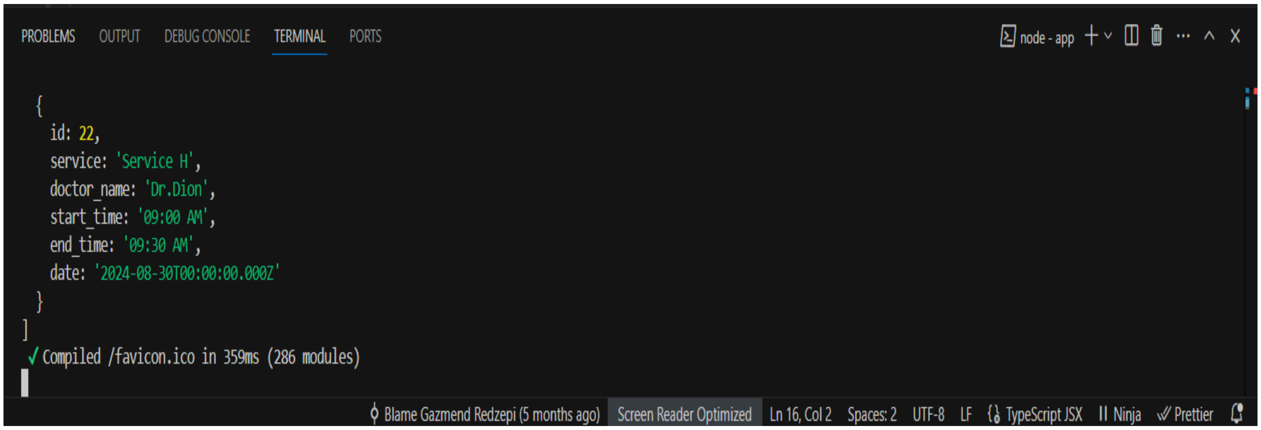
Type Safety: TypeScript is used in the frontend for improved code reliability and maintainability.

API Endpoints

1. Fetch Bookings

- **Endpoint**: `GET /api/bookings`
- **Description**: Retrieves a list of all bookings from the database.
- **Response**: Returns an array of booking objects with fields such as `id`, `service`, `doctor_name`, `start_time`, and `end_time`.

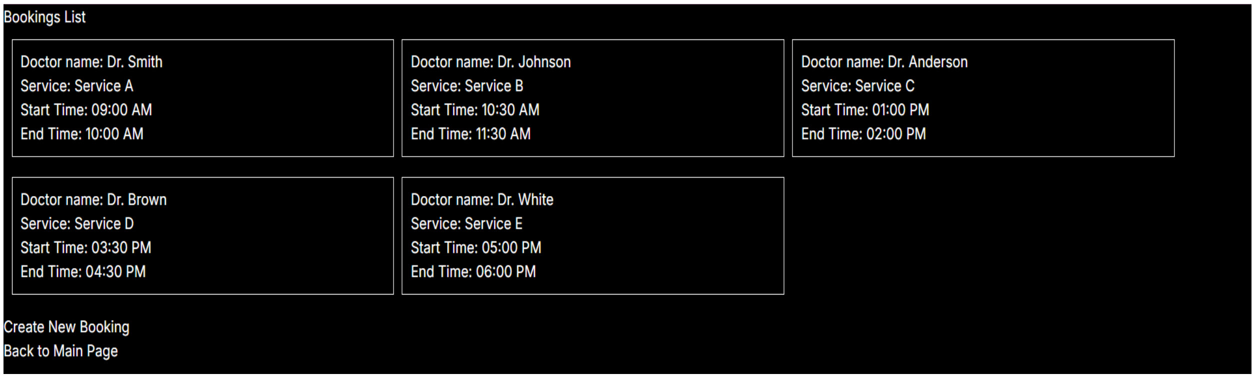
Example Response:



```
  {
    id: 22,
    service: 'Service H',
    doctor_name: 'Dr.Dion',
    start_time: '09:00 AM',
    end_time: '09:30 AM',
    date: '2024-08-30T00:00:00.000Z'
  }
]
✓ Compiled /favicon.ico in 359ms (286 modules)
```

Let's explore my project through images...

1. Upon launching the project, you'll be greeted with the home page, which provides an overview of the available bookings and options to create new ones. The initial view features a clean layout with a list of bookings and navigation links to add new bookings or return to the main page.



Bookings List

Doctor name: Dr. Smith
Service: Service A
Start Time: 09:00 AM
End Time: 10:00 AM

Doctor name: Dr. Johnson
Service: Service B
Start Time: 10:30 AM
End Time: 11:30 AM

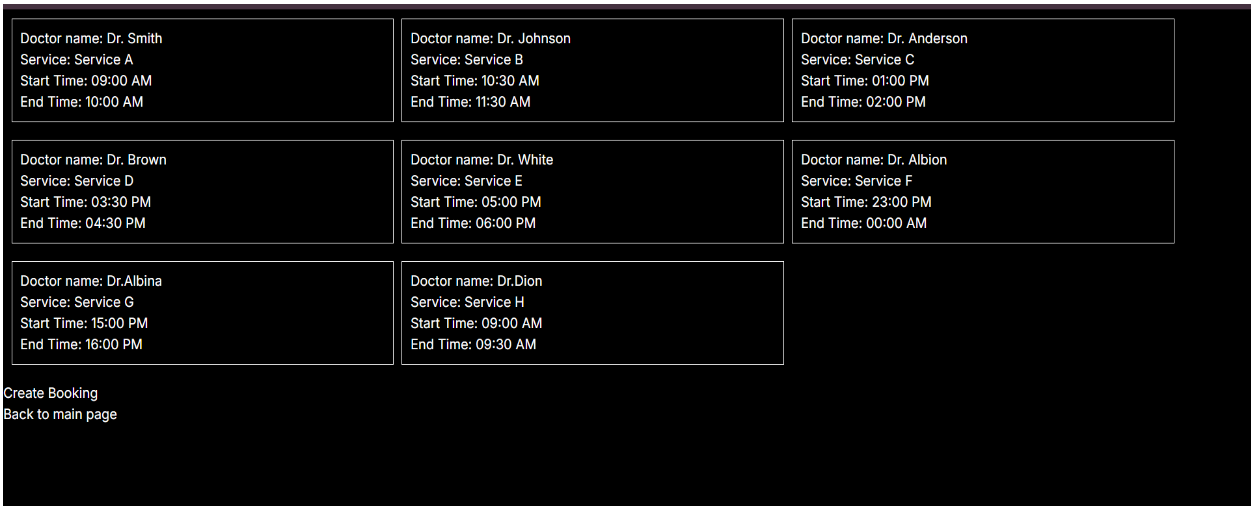
Doctor name: Dr. Anderson
Service: Service C
Start Time: 01:00 PM
End Time: 02:00 PM

Doctor name: Dr. Brown
Service: Service D
Start Time: 03:30 PM
End Time: 04:30 PM

Doctor name: Dr. White
Service: Service E
Start Time: 05:00 PM
End Time: 06:00 PM

Create New Booking
Back to Main Page

2. After clicking 'Back to Main Page,' you will be directed to a page displaying the complete list of bookings. This page provides a clear overview of all available bookings, allowing you to view the details of each entry.



Doctor name: Dr. Smith
Service: Service A
Start Time: 09:00 AM
End Time: 10:00 AM

Doctor name: Dr. Johnson
Service: Service B
Start Time: 10:30 AM
End Time: 11:30 AM

Doctor name: Dr. Anderson
Service: Service C
Start Time: 01:00 PM
End Time: 02:00 PM

Doctor name: Dr. Brown
Service: Service D
Start Time: 03:30 PM
End Time: 04:30 PM

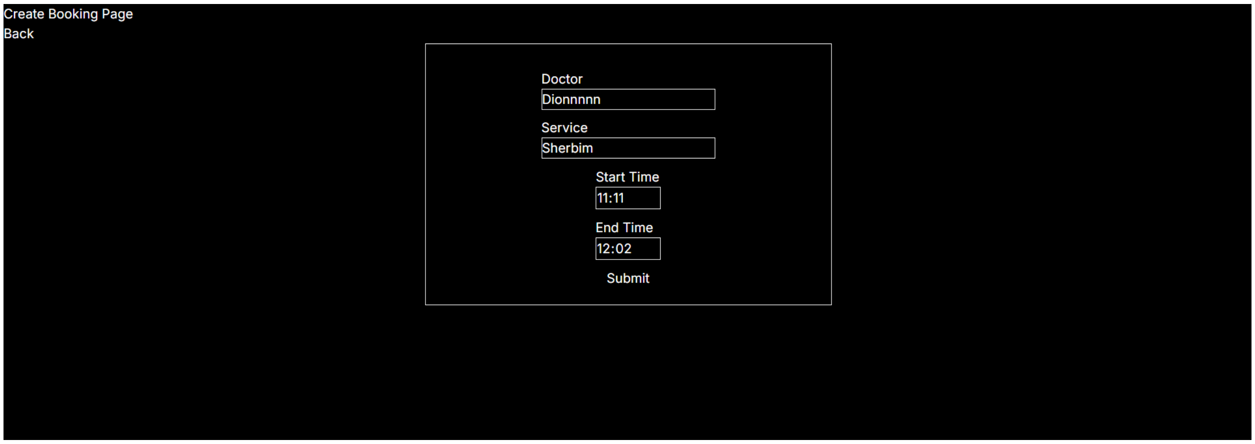
Doctor name: Dr. White
Service: Service E
Start Time: 05:00 PM
End Time: 06:00 PM

Doctor name: Dr. Albion
Service: Service F
Start Time: 23:00 PM
End Time: 00:00 AM

Doctor name: Dr.Albina
Service: Service G
Start Time: 15:00 PM
End Time: 16:00 PM

Doctor name: Dr.Dion
Service: Service H
Start Time: 09:00 AM
End Time: 09:30 AM

3. Clicking 'Create New Booking' will redirect you to a form page where you can enter the details for a new booking. On this page, you'll be able to fill in fields such as service, doctor name, start time and end time then submit the form to add the new booking.



Create Booking Page

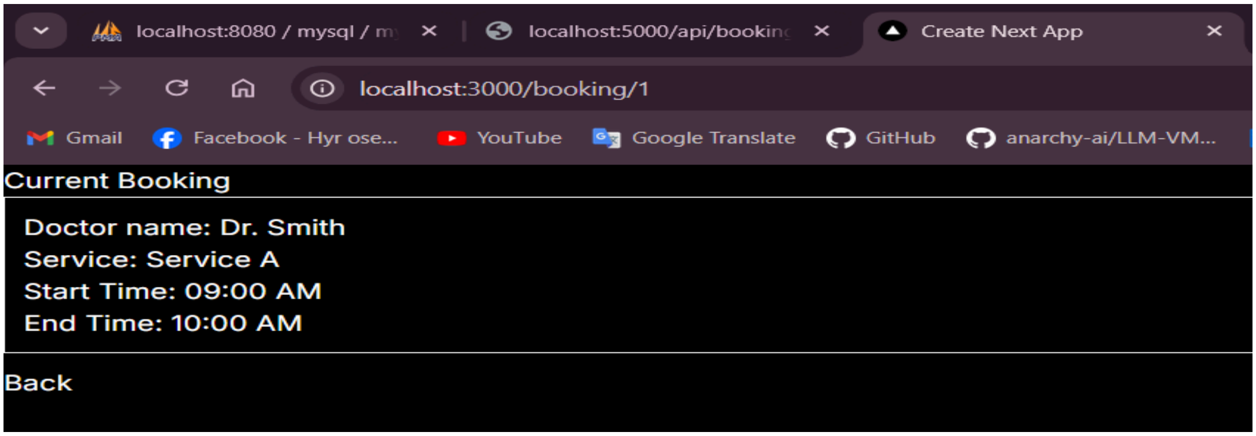
Doctor: Dionnnnn

Service: Sherbim

Start Time: 11:11

End Time: 12:02

4. Clicking on a specific booking will navigate you to a detailed view of that booking. Here, you'll see all the information related to the selected booking, including service, doctor name, start time and end time. You can see the project is running on port 3000 and the booking is number 1.



Current Booking

Doctor name: Dr. Smith
Service: Service A
Start Time: 09:00 AM
End Time: 10:00 AM

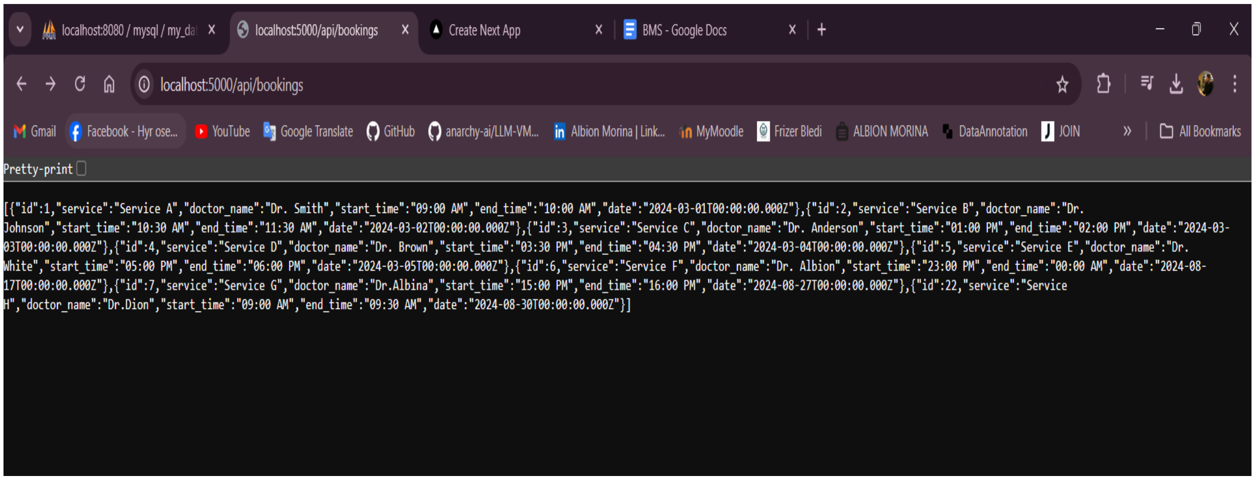
Back

When you visit `http://localhost:5000/api/bookings`, you should see a list of all bookings stored in the database. This API endpoint returns the booking data in JSON format. The data typically includes fields such as:

- `id`: The unique identifier for the booking.
- `service`: The service being booked.
- `doctor_name`: The name of the doctor associated with the booking.
- `start_time`: The start time of the booking.
- `end_time`: The end time of the booking.

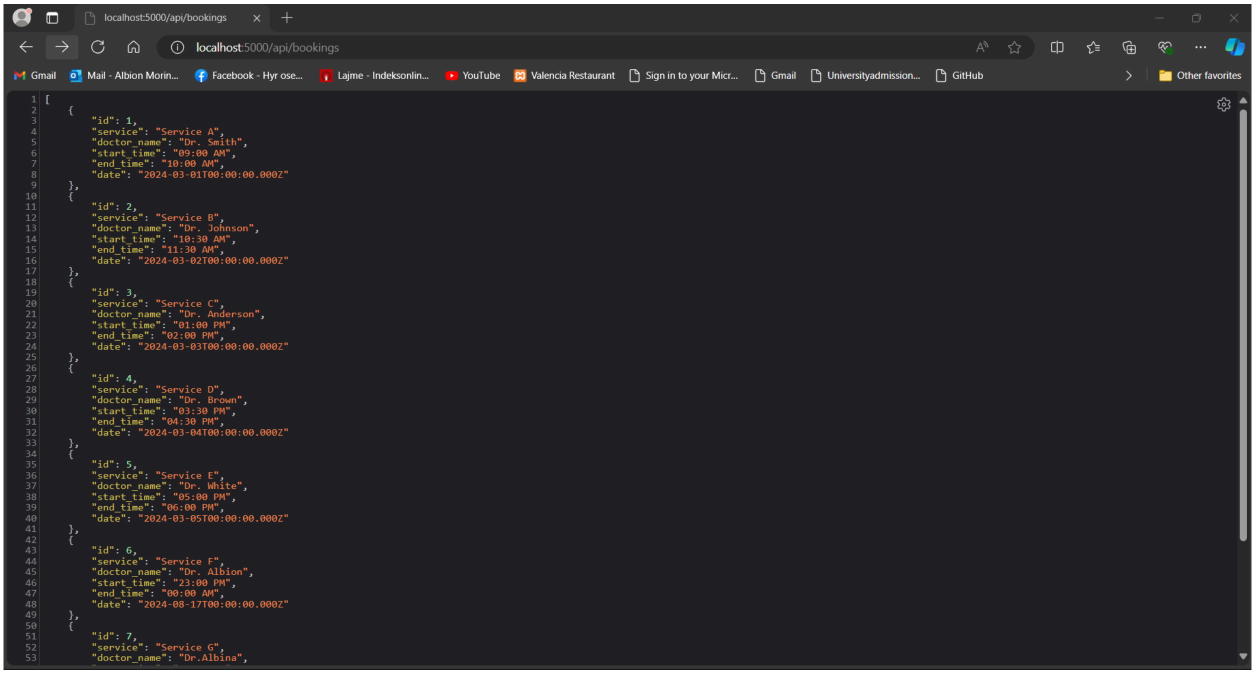
For example, the response might look like this:

Google Chrome:



```
[{"id":1,"service":"Service A","doctor_name":"Dr. Smith","start_time":"09:00 AM","end_time":"10:00 AM","date":"2024-03-01T00:00:00.000Z"},{"id":2,"service":"Service B","doctor_name":"Dr.
Johnson","start_time":"10:30 AM","end_time":"11:30 AM","date":"2024-03-02T00:00:00.000Z"},{"id":3,"service":"Service C","doctor_name":"Dr. Anderson","start_time":"01:00 PM","end_time":"02:00 PM","date":"2024-03-
03T00:00:00.000Z"},{"id":4,"service":"Service D","doctor_name":"Dr. Brown","start_time":"03:30 PM","end_time":"04:30 PM","date":"2024-03-04T00:00:00.000Z"},{"id":5,"service":"Service E","doctor_name":"Dr.
White","start_time":"05:00 PM","end_time":"06:00 PM","date":"2024-03-05T00:00:00.000Z"},{"id":6,"service":"Service F","doctor_name":"Dr. Albion","start_time":"23:00 PM","end_time":"00:00 AM","date":"2024-08-
17T00:00:00.000Z"},{"id":7,"service":"Service G","doctor_name":"Dr.Albina","start_time":"15:00 PM","end_time":"16:00 PM","date":"2024-08-27T00:00:00.000Z"},{"id":22,"service":"Service
H","doctor_name":"Dr.Dion","start_time":"09:00 AM","end_time":"09:30 AM","date":"2024-08-30T00:00:00.000Z"}]
```

Microsoft Edge:

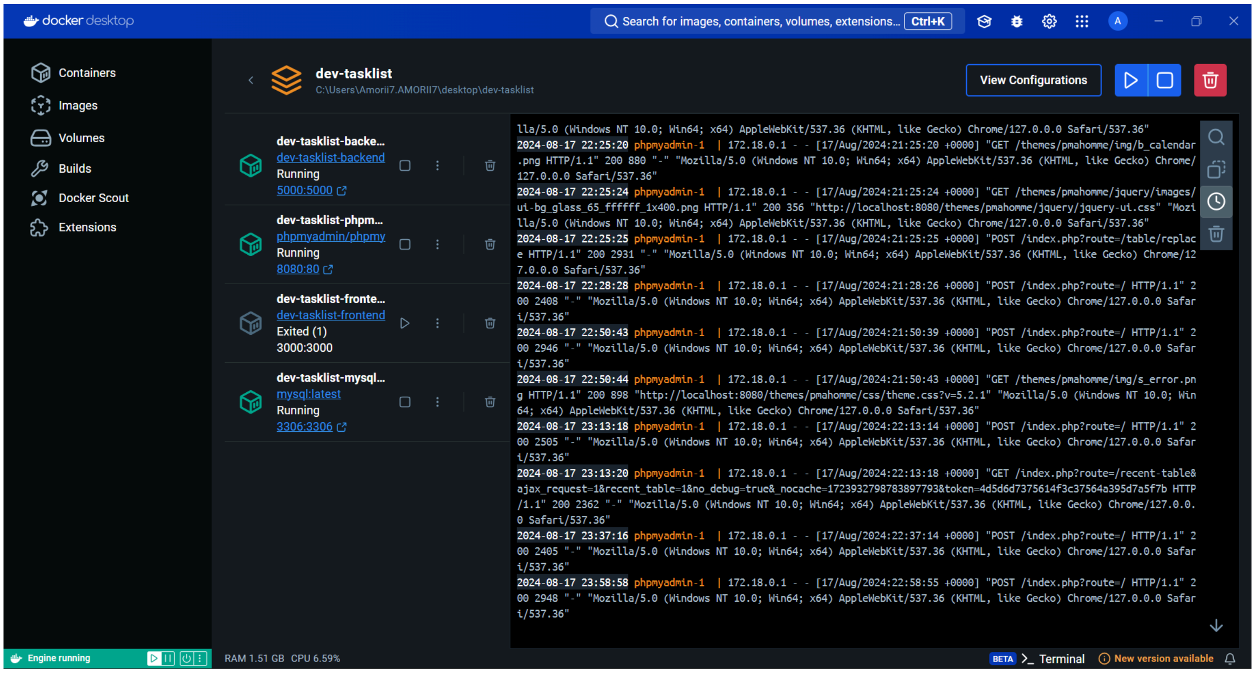


```
[
    {
        "id": 1,
        "service": "Service A",
        "doctor_name": "Dr. Smith",
        "start_time": "09:00 AM",
        "end_time": "10:00 AM",
        "date": "2024-03-01T00:00:00.000Z"
    },
    {
        "id": 2,
        "service": "Service B",
        "doctor_name": "Dr. Johnson",
        "start_time": "10:30 AM",
        "end_time": "11:30 AM",
        "date": "2024-03-02T00:00:00.000Z"
    },
    {
        "id": 3,
        "service": "Service C",
        "doctor_name": "Dr. Anderson",
        "start_time": "01:00 PM",
        "end_time": "02:00 PM",
        "date": "2024-03-03T00:00:00.000Z"
    },
    {
        "id": 4,
        "service": "Service D",
        "doctor_name": "Dr. Brown",
        "start_time": "03:30 PM",
        "end_time": "04:30 PM",
        "date": "2024-03-04T00:00:00.000Z"
    },
    {
        "id": 5,
        "service": "Service E",
        "doctor_name": "Dr. White",
        "start_time": "05:00 PM",
        "end_time": "06:00 PM",
        "date": "2024-03-05T00:00:00.000Z"
    },
    {
        "id": 6,
        "service": "Service F",
        "doctor_name": "Dr. Albion",
        "start_time": "23:00 PM",
        "end_time": "00:00 AM",
        "date": "2024-08-17T00:00:00.000Z"
    },
    {
        "id": 7,
        "service": "Service G",
        "doctor_name": "Dr.Albina",
```

Docker Overview

Docker Dashboard Overview: "The Docker Dashboard provides a comprehensive view of running containers, including their statuses, resource usage, and logs. With docker you can manage and monitor your Docker environment directly from this interface."

MY DOCKER CONTAINER VIEW: *"In my Docker Container View, you can see the list of all running and stopped containers, their current status, and their resource usage metrics."*



dev-tasklist
C:\Users\Amorii7.AMORII7\desktop\dev-tasklist

dev-tasklist-backe…
dev-tasklist-backend
Running
5000:5000

dev-tasklist-phpm…
phpmyadmin/phpmy
Running
8080:80

dev-tasklist-fronte…
dev-tasklist-frontend
Exited (1)
3000:3000

dev-tasklist-mysql…
mysql:latest
Running
3306:3306

```
lla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36"
2024-08-17 22:25:20 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:21:25:20 +0000] "GET /themes/pmahomme/img/b_calendar
.png HTTP/1.1" 200 880 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
127.0.0.0 Safari/537.36"
2024-08-17 22:25:24 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:21:25:24 +0000] "GET /themes/pmahomme/jquery/images/
ui-bg_glass_65_ffffff_1x400.png HTTP/1.1" 200 356 "http://localhost:8080/themes/pmahomme/jquery/jquery-ui.css" "Mozi
lla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36"
2024-08-17 22:25:25 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:21:25:25 +0000] "POST /index.php?route=/table/replac
e HTTP/1.1" 200 2931 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/12
7.0.0.0 Safari/537.36"
2024-08-17 22:28:28 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:21:28:26 +0000] "POST /index.php?route=/ HTTP/1.1" 2
00 2408 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safar
i/537.36"
2024-08-17 22:50:43 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:21:50:39 +0000] "POST /index.php?route=/ HTTP/1.1" 2
00 2946 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safar
i/537.36"
2024-08-17 22:50:44 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:21:50:43 +0000] "GET /themes/pmahomme/img/s_error.pn
g HTTP/1.1" 200 898 "http://localhost:8080/themes/pmahomme/css/theme.css?v=5.2.1" "Mozilla/5.0 (Windows NT 10.0; Win
64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safari/537.36"
2024-08-17 23:13:18 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:22:13:14 +0000] "POST /index.php?route=/ HTTP/1.1" 2
00 2505 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safar
i/537.36"
2024-08-17 23:13:20 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:22:13:18 +0000] "GET /index.php?route=/recent-table&
ajax_request=1&recent_table=1&no_debug=true&_nocache=1723932798783897793&token=4d5d6d7375614f3c37564a395d7a5f7b HTTP
/1.1" 200 2362 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.
0 Safari/537.36"
2024-08-17 23:37:16 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:22:37:14 +0000] "POST /index.php?route=/ HTTP/1.1" 2
00 2405 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safar
i/537.36"
2024-08-17 23:58:58 phpmyadmin-1  | 172.18.0.1 - - [17/Aug/2024:22:58:55 +0000] "POST /index.php?route=/ HTTP/1.1" 2
00 2948 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.0.0 Safar
i/537.36"
```

References

[1] https://nextjs.org/docs/getting-started/installation

[2] https://stackoverflow.com/

[3] https://blog.postman.com/

[4] https://chatgpt.com/

Conclusion

This project provides a comprehensive solution for managing bookings through a web application. By leveraging Next.js, Express.js, and MySQL, the application facilitates booking creation, viewing, and management with an intuitive user interface. The Docker setup ensures a streamlined development and deployment process, with all components integrated seamlessly. With clear API endpoints and a user-friendly design, this project demonstrates effective use of modern web technologies to build and manage a functional booking system.