

Visitor hostel module

Student mentor : Deepanshu kumar(21bcs072)

TEAM MEMBERS :

Sparsh Ranjan(21bcs205)

Varun Raj(21bcs236)

Avneesh Dayal(21bcs046)

Azemera Vishnu Nayak(21bcs051)

Sahil Chauksey(21bcs181)

All the API used in the module below :

1. Methods for booking
2. Check Status
3. Getting booking form
4. Request booking
5. Update booking
6. Confirm booking by VH Incharge
7. Cancel booking by VH Incharge
8. Cancel confirmed booking by intender
9. Cancel booking request
10. Reject Booking request
11. Guest check in view
12. Guest check out view

- 13. Record meal
- 14. Generate bill
- 15. Room availability Details
- 16. Add to Inventory details
- 17. Booking details

Overview of the module :

According to the overview by our team we found that the basic login credentials of the global API is missing also we cannot check the drilling API.

Also the above explained 17 API have some potential errors like 404 error , forbidden access and dependencies problem . We have to make many changes into the API and some API are very poor so we have to write from scratch also due to the dependencies with the database the earlier team has created so many databases for the single work so have to make them correct also .

APIs :

Already Implemented API

- **Req_for_booking:**

The Intender will be able to book room(s) in the VH. The Intender has to fill the form in the required format and submit for further processing.

- Database - **get_booking_requests**

■

```

Status: 200 OK   Size: 0 Bytes   Time: 188 ms

Response  Headers 10  Cookies 1  Results  Docs  {}  ≡

1  {
2    "form": {
3      "fields": {
4        "login": {
5          "label": "Login",
6          "value": null,
7          "help_text": "",
8          "errors": [],
9          "widget": {
10           "attrs": {
11             "placeholder": "Username or e-mail",
12             "autocomplete": "email"
13           }
14         }
15       },
16       "password": {
17         "label": "Password",
18         "value": null,
19         "help_text": "",
20         "errors": []
21       }
22     }
23   }

```

Copy

Response Chart ↗

← → ↻ 127.0.0.1:8000/visitorhostel/ 🔍 ☆ 🌐 📱 🖨️ 👤 ⋮

college email canvas YouTube Home - Chess.com Practice gpt Elective Selection Section B_AI_2023 - G... Btech 3rd year previo... All Bookmarks

≡ 🏠 Dashboard 🗨️ Feedback

VARUN RAJ 📶 Log

Place a Request

Booking Details Visitor's Details

Arrival Date *

From

Arrival Time *: Hours

Select

Minutes *

Select

AM / PM *

Select

Departure Date *

To

Departure Time *: Hours

Select

Minutes *

Select

AM / PM *

Select

Number Of People *

1

Number Of Rooms *

1

Category *

Select

Purpose Of Visit *

Remarks

Next

VARUN RAJ CSE

B.Tech 2021

Sem - 3

Manage Bookings >

Booking Form >

Rules and Regulations >

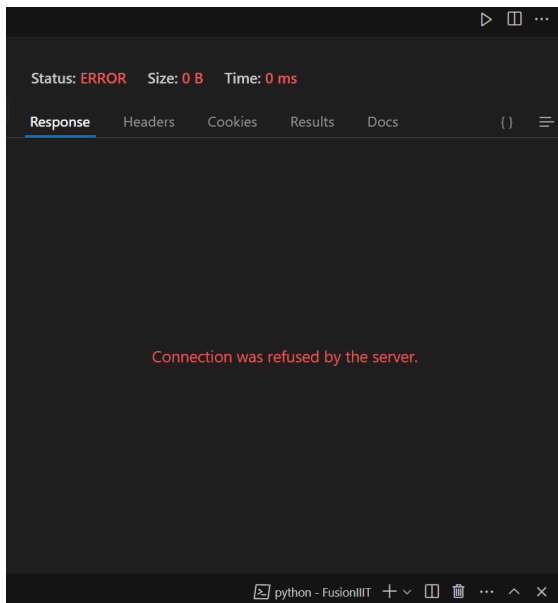
API status : working

Partially implemented API's:

- **Check_status :**

After submitting a booking request, the room Intender can check if the booking is confirmed/pending/canceled.

- Database - No table created

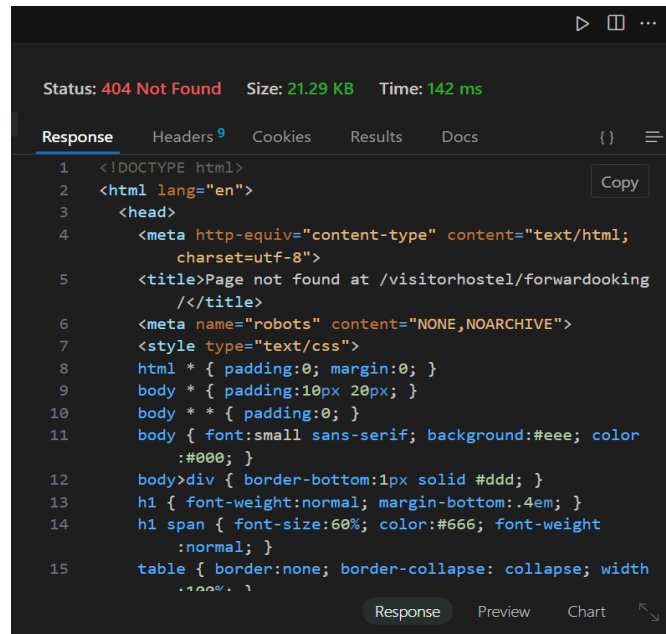


Status code: 500(Internal Server Error)

- **Cancel_booking :**

Intenders will be able to cancel their booking of room(s) in the VH with remarks.

- Database - No table created



The screenshot shows a web browser's developer tools interface. At the top, the status bar indicates 'Status: 404 Not Found', 'Size: 21.29 KB', and 'Time: 142 ms'. Below this, the 'Response' tab is selected, displaying the HTML content of the response. The HTML starts with a DOCTYPE declaration, followed by an HTML tag with lang='en'. The head section contains a meta tag for content-type, a title 'Page not found at /visitorhostel/forwardlooking', and a robots meta tag. The body contains a CSS style block and several HTML elements including a div, h1, h1 span, and a table. A 'Copy' button is visible next to the first line of the HTML. At the bottom of the response pane, there are buttons for 'Response', 'Preview', and 'Chart'.

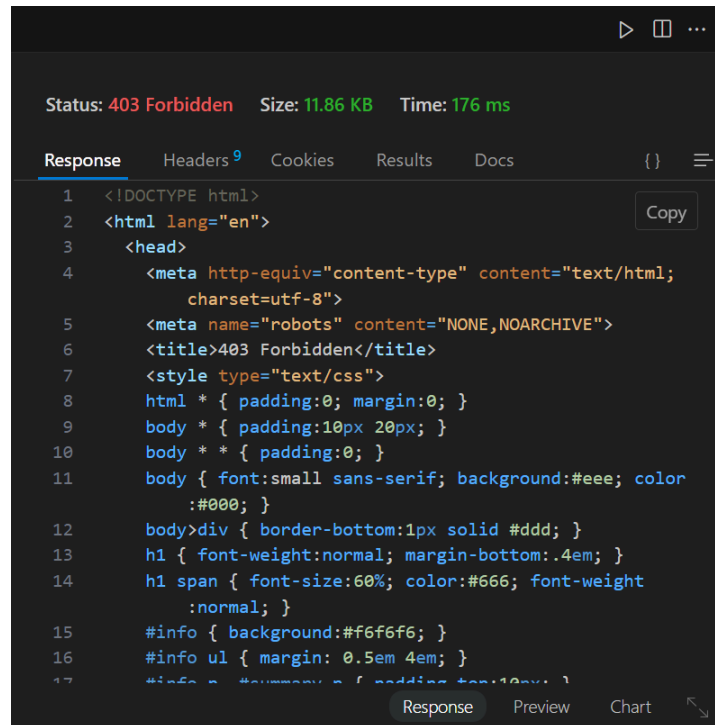
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta http-equiv="content-type" content="text/html;
      charset=utf-8">
5     <title>Page not found at /visitorhostel/forwardlooking
      /</title>
6     <meta name="robots" content="NONE,NOARCHIVE">
7     <style type="text/css">
8       html * { padding:0; margin:0; }
9       body * { padding:10px 20px; }
10      body * * { padding:0; }
11      body { font:small sans-serif; background:#eee; color
        :#000; }
12      body>div { border-bottom:1px solid #ddd; }
13      h1 { font-weight:normal; margin-bottom:.4em; }
14      h1 span { font-size:60%; color:#666; font-weight
        :normal; }
15      table { border:none; border-collapse: collapse; width
        :100%; }
```

Status code: 404(Not Found)

- **Check_availability :**

The Vh caretaker will be able to check the current occupancy of VH.

- Database - No table created



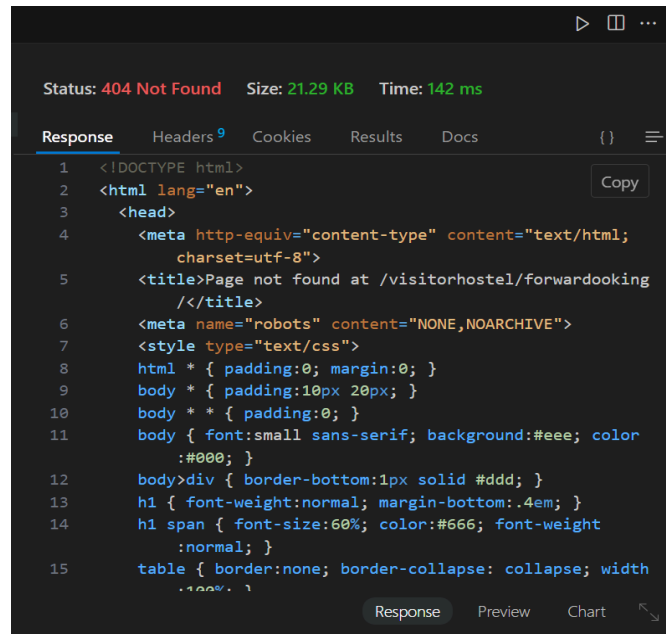
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5     <meta name="robots" content="NONE,NOARCHIVE">
6     <title>403 Forbidden</title>
7     <style type="text/css">
8       html * { padding:0; margin:0; }
9       body * { padding:10px 20px; }
10      body * * { padding:0; }
11      body { font:small sans-serif; background:#eee; color
12             :#000; }
13      body>div { border-bottom:1px solid #ddd; }
14      h1 { font-weight:normal; margin-bottom:.4em; }
15      h1 span { font-size:60%; color:#666; font-weight
16               :normal; }
17      #info { background:#f6f6f6; }
18      #info ul { margin: 0.5em 4em; }
19      #info p { background:#f6f6f6; padding-top:10px; }
```

Status code: 403(Forbidden)

- **Allot_rooms :**

The caretaker will allot rooms according to the availability .

- **Database - No table created**



The screenshot shows a web browser's developer tools interface. At the top, the status bar indicates 'Status: 404 Not Found', 'Size: 21.29 KB', and 'Time: 142 ms'. Below this, the 'Response' tab is selected, displaying the HTML content of the response. The HTML starts with a DOCTYPE declaration, followed by an HTML tag with lang='en'. The head section includes a meta tag for content type (text/html; charset=utf-8), a title 'Page not found at /visitorhostel/forwardooking', and a robots meta tag. The body contains a CSS style block and a table. A 'Copy' button is visible next to the first line of the HTML. At the bottom of the response pane, there are buttons for 'Response', 'Preview', and 'Chart'.

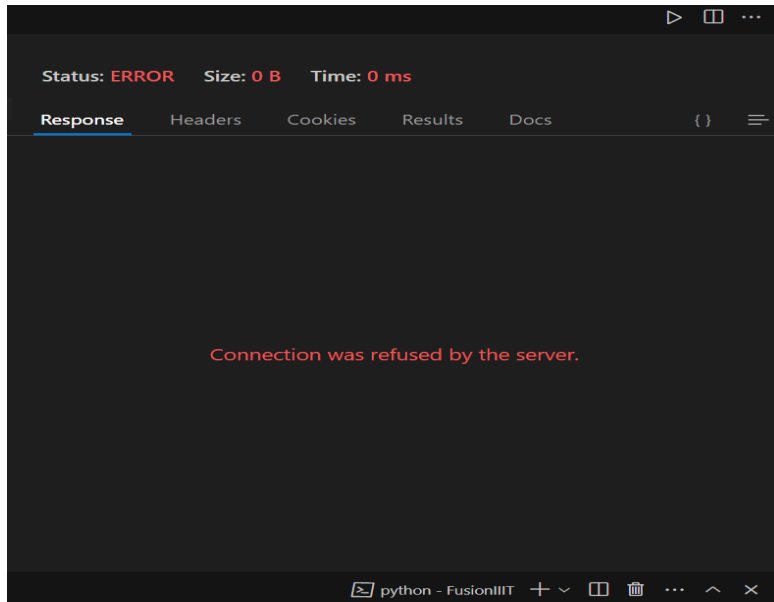
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta http-equiv="content-type" content="text/html;
      charset=utf-8">
5     <title>Page not found at /visitorhostel/forwardooking
      /</title>
6     <meta name="robots" content="NONE,NOARCHIVE">
7     <style type="text/css">
8       html * { padding:0; margin:0; }
9       body * { padding:10px 20px; }
10      body * * { padding:0; }
11      body { font:small sans-serif; background:#eee; color
        :#000; }
12      body>div { border-bottom:1px solid #ddd; }
13      h1 { font-weight:normal; margin-bottom:.4em; }
14      h1 span { font-size:60%; color:#666; font-weight
        :normal; }
15      table { border:none; border-collapse: collapse; width
        :100%; }
```

Status code: 404(Not Found)

- **Cancel_booking :**

VH Caretaker will be able to cancel any booking of room(s) in VH with remarks.

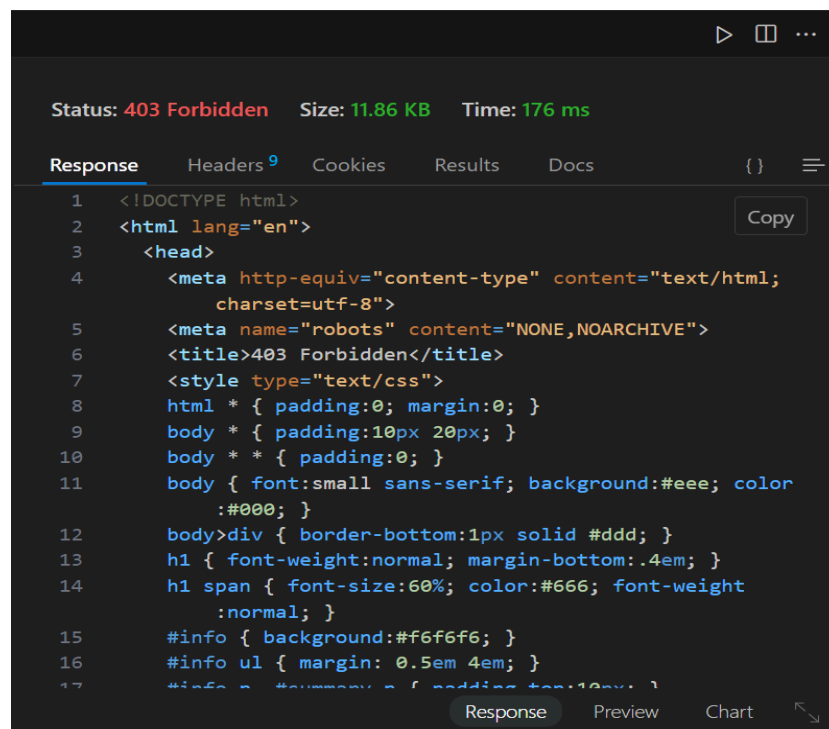
- Database - No table created



Status code: 500(Internal Server Error)

- **Manage_guest_check_in :**

VH Caretaker will be able to mark the check in status of a booked VH room.

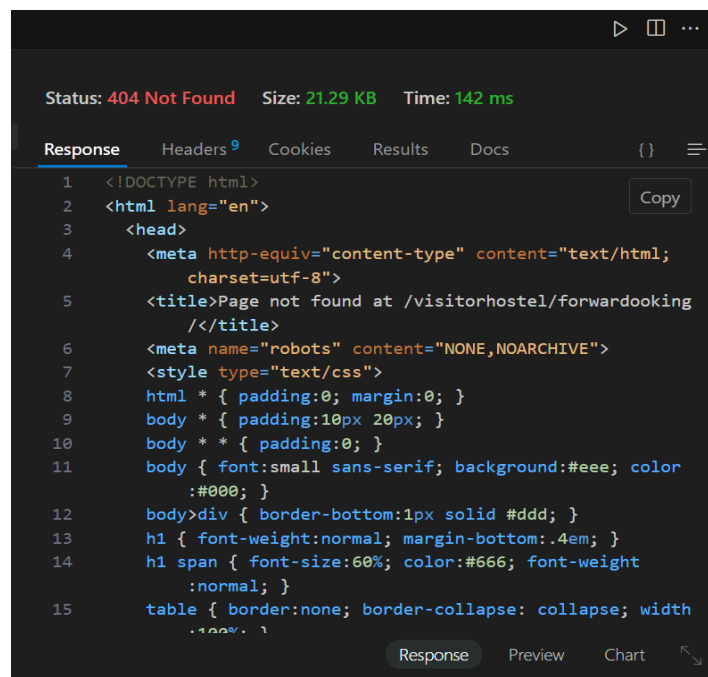


Status code: 403(Forbidden)

- **Manage_guest_check_out**

VH Caretaker will be able to initiate the process of checkout on the system regarding bill clearance and room status update.

- Database - No table created



The screenshot shows a web browser's developer tools interface. At the top, the status is '404 Not Found' in red, with 'Size: 21.29 KB' and 'Time: 142 ms' in green. Below this, the 'Response' tab is selected, displaying the HTML content of the error page. The HTML includes a DOCTYPE declaration, a head section with meta tags for content type, charset, and robots, and a style block with CSS rules for padding, margin, font, and background. The body contains a div with a border-bottom and an h1 element with a large span.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=utf-8">
5     <title>Page not found at /visitorhostel/forwardlooking
6     <meta name="robots" content="NONE,NOARCHIVE">
7     <style type="text/css">
8       html * { padding:0; margin:0; }
9       body * { padding:10px 20px; }
10      body * * { padding:0; }
11      body { font:small sans-serif; background:#eee; color
12            :#000; }
13      body>div { border-bottom:1px solid #ddd; }
14      h1 { font-weight:normal; margin-bottom:.4em; }
15      h1 span { font-size:60%; color:#666; font-weight
16            :normal; }
17      table { border:none; border-collapse: collapse; width
18            :100%; }
```

Status code: 404(Not Found)

API's yet to be implemented by scratch:

- **manage_noTurnUp :**

The system will notify VH Caretaker/VH In-charge about guests who didn't turn up. The VH Caretaker can take actions as deemed fit.

- **Manage_bill_settlement :**

VH Caretaker sees if the bill is paid by intender/guest. He collects the money according to the bill and updates the system accordingly.

- **Manage_inventory :**

VH Caretaker will be able to manage the inventory of the VH.

- **Get_low_on_stock_alerts :**

VH Caretaker will get alerts when any inventory item quantity falls below a pre-decided threshold.

- **Update_consumption :**

The VH Caretaker will periodically update the system with the quantity of items consumed.

- **View_bookings :**

The VH In-charge will be able to check the booking history and current occupancy of VH.

- **Confirm_booking :**

VH In-Charge will grant the confirmation of the booking after verifying the information provided in the application

- **Reject_booking :**

VH In-charge will be able to reject any booking of room(s) in VH with remarks

- **Current problems facing with the module or in its use cases:-**

The main problems facing the booking module are:

Existing API issues:

- Missing login credentials: Cannot access or verify the global API.
- Drilling API inaccessible: Limits functionality and data analysis.
- Error-prone APIs: Frequent 400, 403, and 500 errors indicate instability and potential data inconsistencies.
- Database redundancy: Multiple databases for single tasks complicate maintenance and increase error risk.

Incomplete APIs:

- Partially implemented APIs: Missing functionality in Check_status, Cancel_booking, Check_availability, Allot_rooms, and Manage_guest_check_in, Manage_guest_check_out APIs hinder workflow.

Missing APIs:

- Critical functionality gaps: Lack of manage_noTurnUp, Manage_bill_settlement, Manage_inventory, Get_low_on_stock_alerts, Update_consumption, and View_bookings APIs limit caretaker and in-charge capabilities.

Overall, the module suffers from inadequate security, unstable functionality, and incomplete features, significantly impacting its usability and reliability.