

Droame Web Application

Technologies Used:

Frontend

- HTML
- Java Script
- Boot Strap
- CSS

Backend

- Python 3
- My SQL
- FastAPI
- Uvicorn
- JWT
- Pydantic
- Sqlalchemy
- passlib

Tools

- Git
- GitHub
- VS Code
- Thunder Client

How to Run (Droame) Web-Application

- [Clone this repo](#)
- Setup Python Virtual Environment(not necessary)
- Install all [requirement](#) from requirements.txt file
- Run uvicorn app.main:app --reload or uvicorn app.main:app to start API services
- Register operator by using operator signup API
- Run any live server to host index.html web page
- Use the Operator id which use signup earlier to login
- After redirecting on operator.html page you can perform operator operations

Signup API body (JSON)

<http://127.0.0.1:8000/signup>

```
{  
    "operator_id": "operator_2",  
    "password": "op2"  
}
```

#index.html Page Flow

(note: There are 3 API for operator login, registration)

For Registration :-

- **Signup API** (<http://127.0.0.1:8000/signup>)

#JSON BODY

```
{  
  
  "operator_id": "operator_2",  
  
  "password": "op2"  
  
}
```

For Login :-

- **Signin API** - (<http://127.0.0.1:8000/signin>)

#JSON BODY

```
{  
  
  "operator_id": "operator_2",  
  
  "password": "op2"  
  
}
```

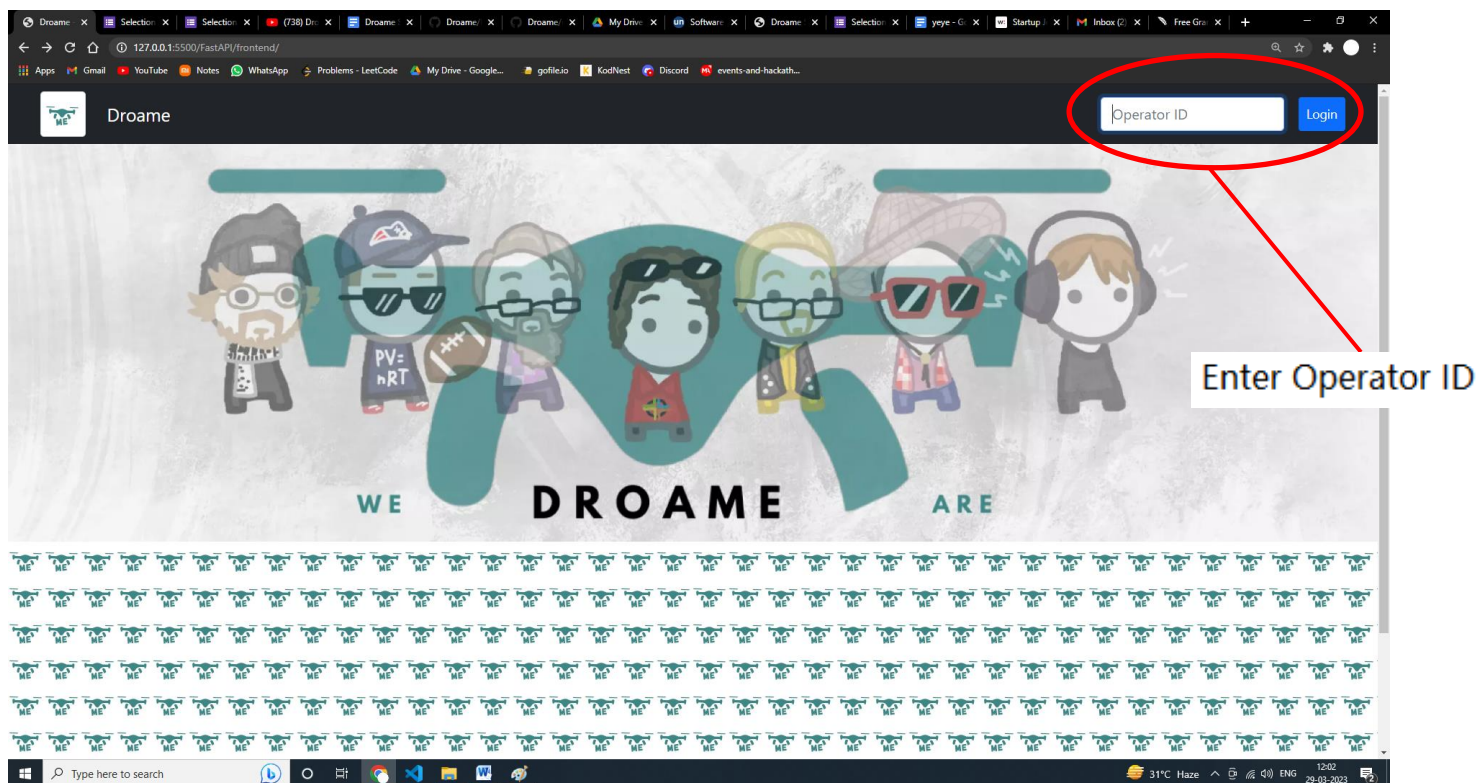
- **Login API** - (<http://127.0.0.1:8000/login>)

```
{  
  
  "operator_id": "operator_2"  
  
}
```

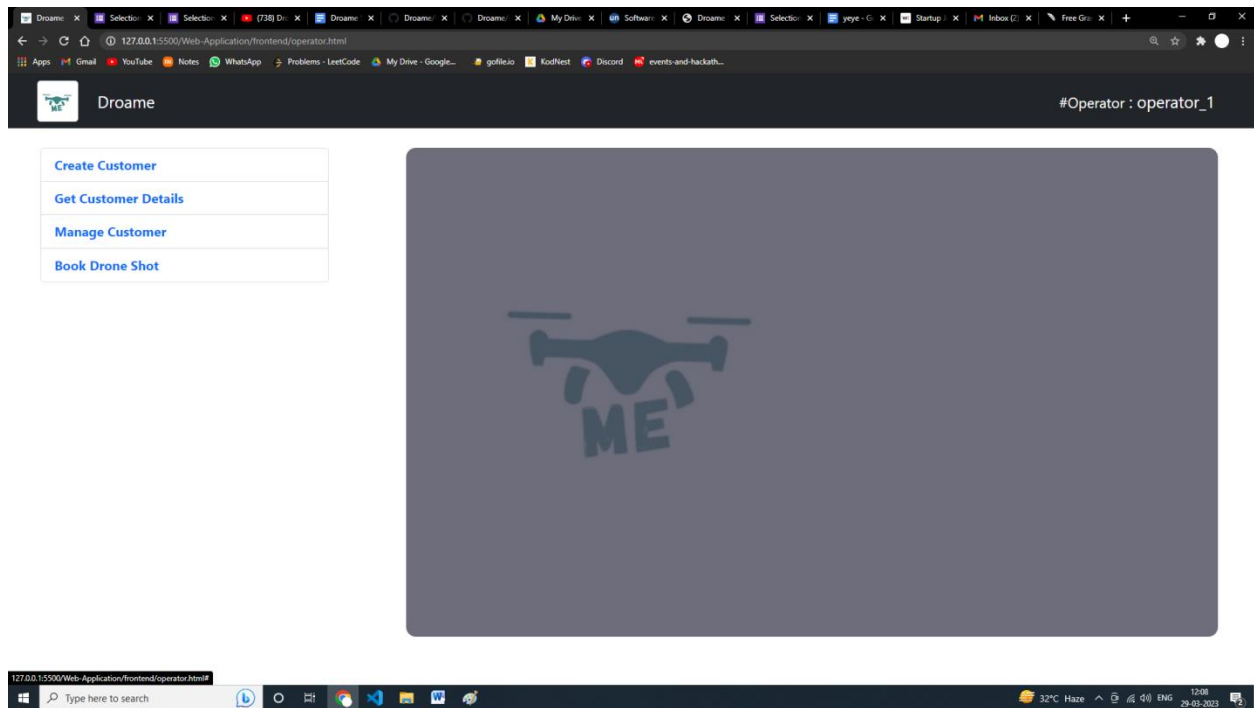
(Note: To login As Operator your need to register operator first by using Signup API)

After Operator registration Operator can login by using opeartor

Only login API is in use for login to as operator (other two api are just for additional if we required them in future Signin api required Password)



operator.html Page Flow

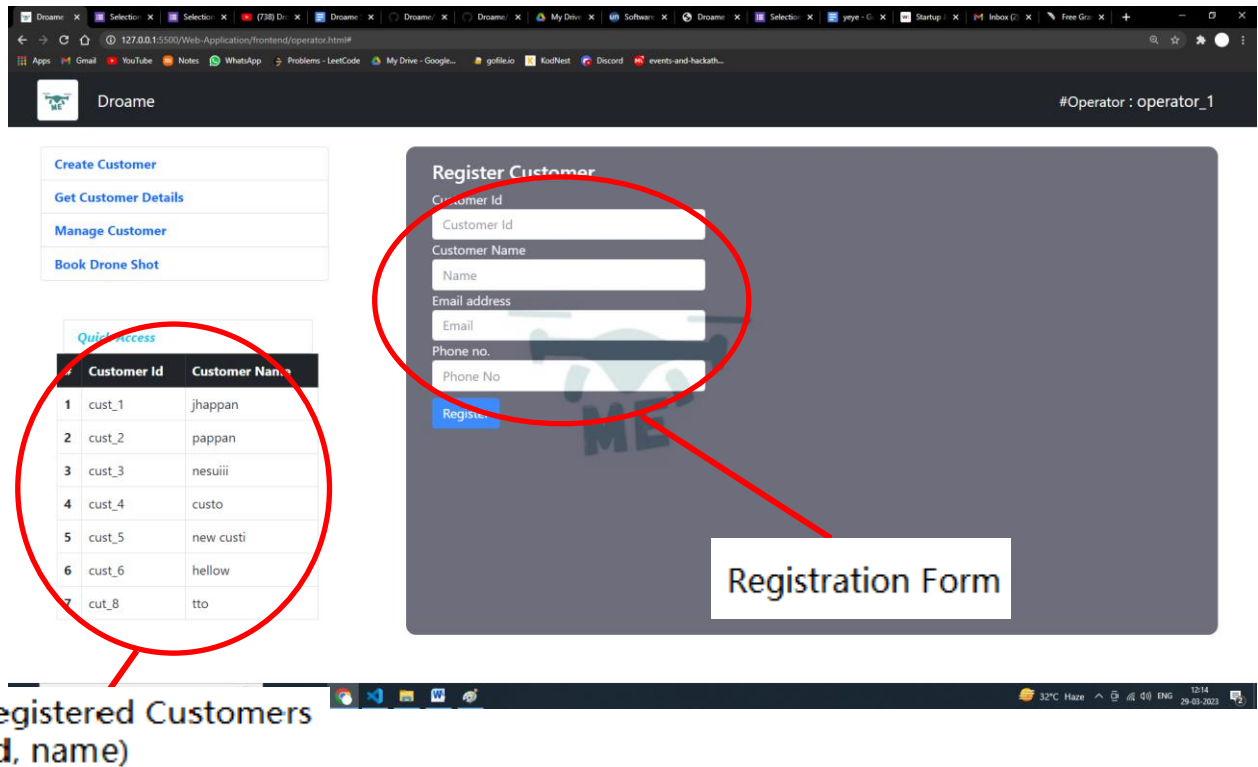


Operator Page have following features and operations

- Creater Customer - To register the customer.
- Get Customer Details – To fetch customer details by using customer Id.
- Manage Customer Details - To delete cusotmer or update customer details like(name, email, phone no).
- Book Drone Shot – To book Drone shot

Breif Options

>> *Creator Customer*



Registration Form

->> It is connected to API to register the customer in database (**API** : <http://127.0.0.1:8000/register>)

Details required to register new Customer are follow (note : * are Unique)

- Customer Id *
- Name
- Email id*
- Phone No*

It will collect data from form and through register API POST request transfer the data

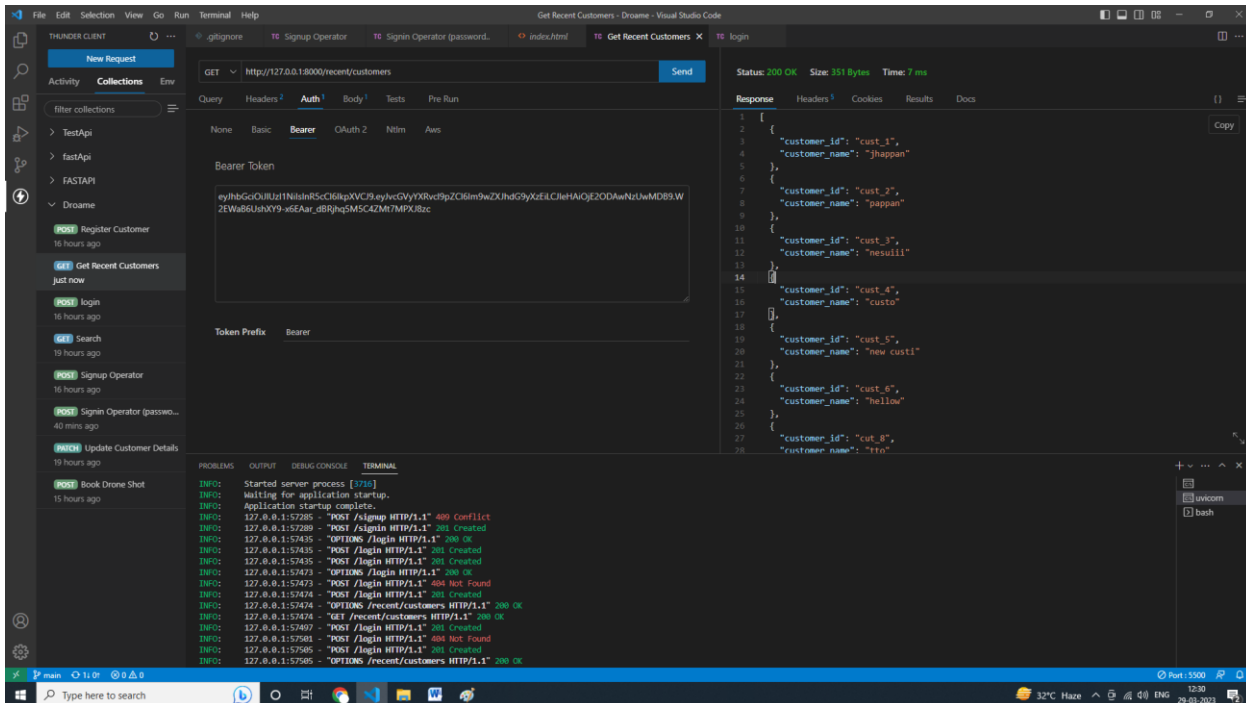
#JSON BODY

```
{  
  "operator_id": "operator_1",  
  "customer_id": "cust_1",  
  "customer_name": "custoo",  
  "customer_email": "custoo@droame.com",  
  "phone_number": "+312 3458456451"  
}
```

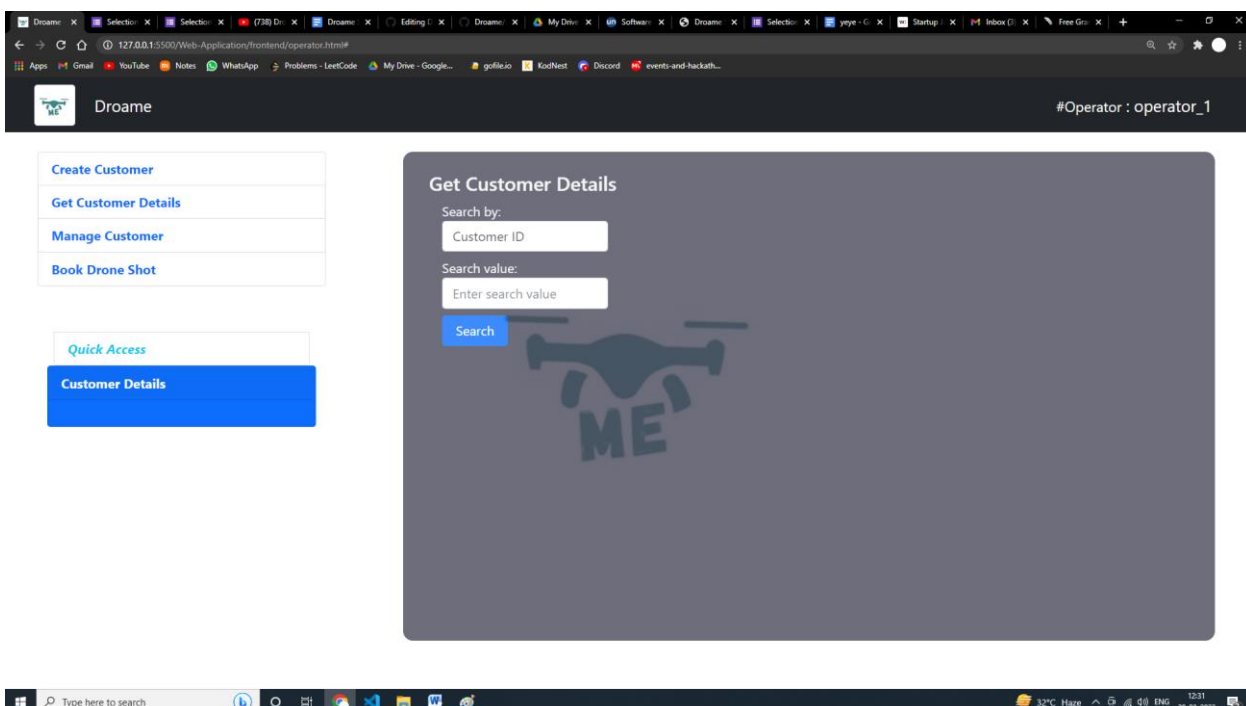
Quick Access (left side to display customers)

->> It is fetching data from recent customer (**API** : <http://127.0.0.1:8000/recent/customers>)

It will return and through GET request



>> Get Customer Details



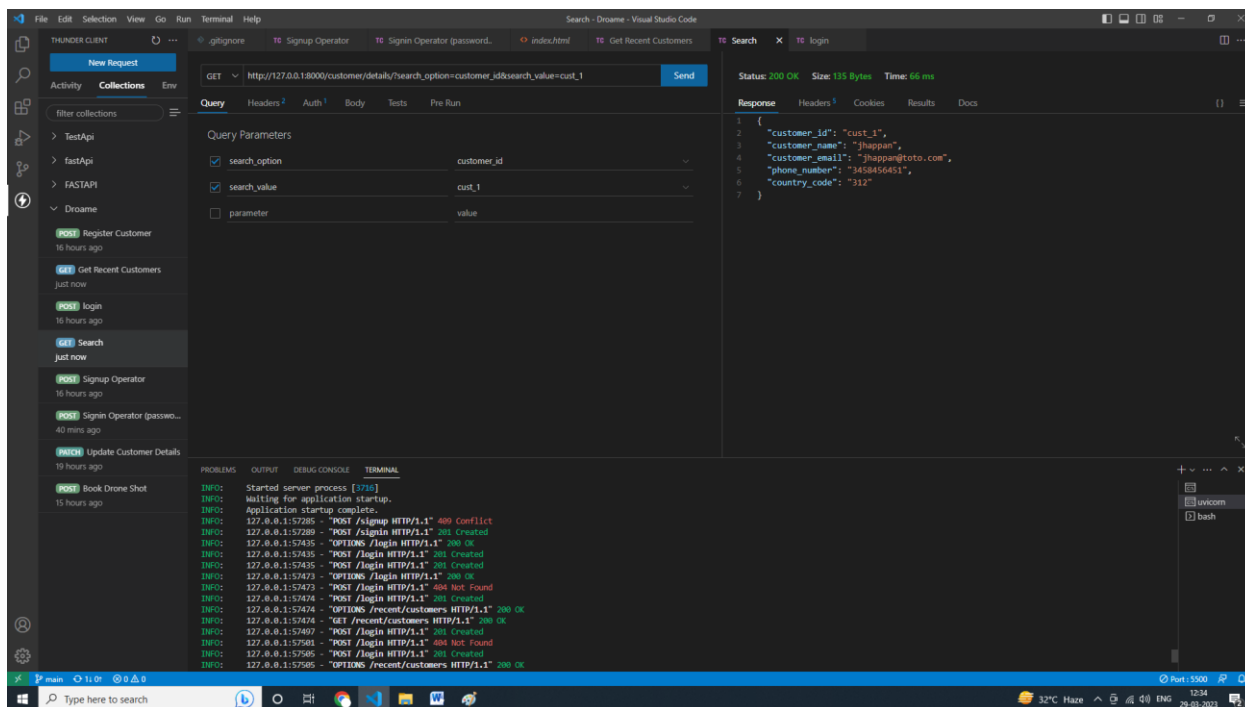
Request Form

->> It is connected to get Details (API :

http://127.0.0.1:8000/customer/details/?search_option={search_option}&search_value={search_value})

\$value1 is search options like : customer_id, customer_email, phone_number

\$value2 is value entered by operator



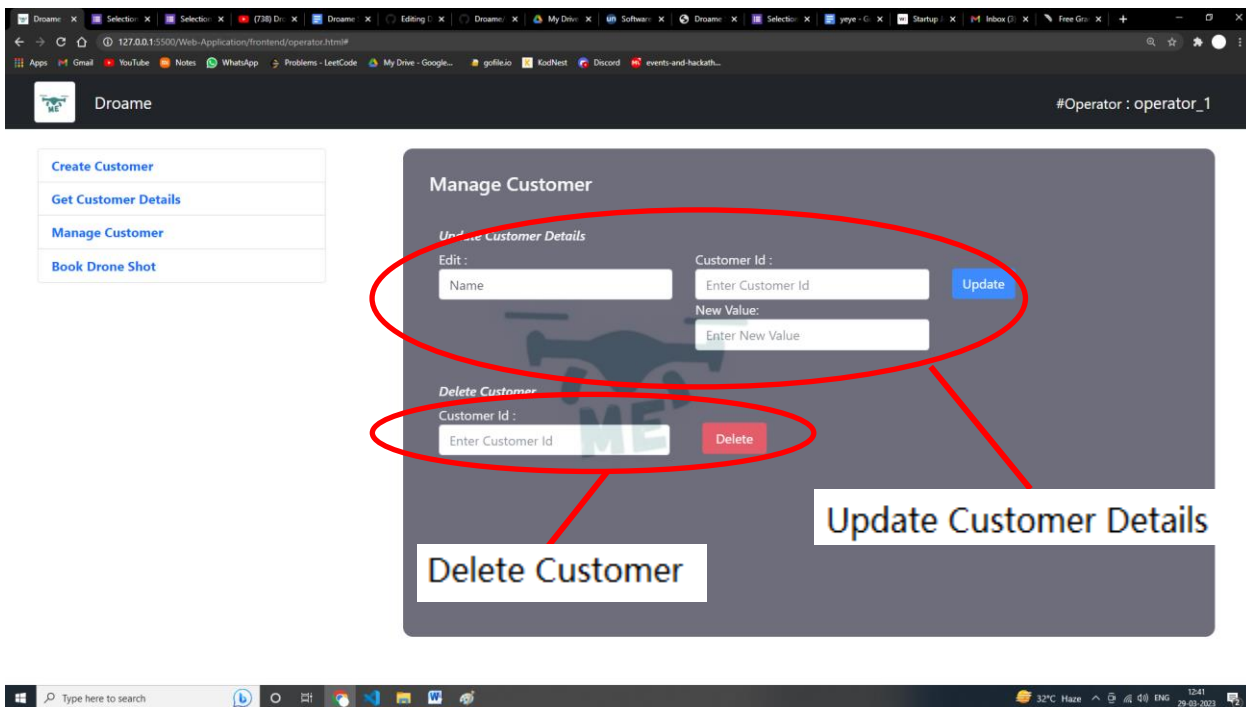
It is GET request API

Return

#JSON BODY

```
{  
  
  "customer_id": "cust_1",  
  
  "customer_name": "jhappan",  
  
  "customer_email": "jhappan@toto.com",  
  
  "phone_number": "3458456451",  
  
  "country_code": "312"  
}
```


>> Manage Customer



->>Update Customer Details

- Selection the detail which you want to update from Edit dropdown
- Enter Customer Id
- Enter New Value

On Clicking On Update Button PATCH request API will call

(API : http://127.0.0.1:8000/update/{customer_id})

\$customer_id is customer_id

#JSON BODY

```
{  
  "customer_name": "jhappan",  
  "customer_email": "jhappan@toto.com",  
  "phone_number": "3458456451"  
}
```

(note : this api can update one detail at one time)

["customer_name", "customer_email", "phone_number"]

->>Delete Customer

Operator can remove its customers from data base using delete API

(API : http://127.0.0.1:8000/delete/{customer_id})

\$customer_id is customer_id

>> Book Drone Shot

The screenshot shows a web browser window with the 'Droame' application. On the left, a sidebar menu includes 'Create Customer', 'Get Customer Details', 'Manage Customer', and 'Book Drone Shot'. The 'Book Drone Shot' option is highlighted. A 'Quick Access' popup is visible, showing 'Booking Details' for a booking with ID 25, customer ID cust_5, location New York, and event time 2023-03-31T04:53. The main content area displays the 'Create Drone Shot' form, which includes fields for Customer ID (cust_5), Location (New York), Date and Time (31-03-2023 04:53), and Drone Shot Type (180-degree shot). A 'Book' button is present. A red circle highlights the form fields, and a red arrow points from the 'Booking Details' popup to the form. A label 'Booking Form' points to the form, and a label 'Booking Details' points to the popup.

->>Drone Booking Form

Operator can book the drone shot using this form. This form is connected to booking API : (<http://127.0.0.1:8000/booking>)

#JSON BODY

```
{  
  "customer_id": "cust_2",  
  "booked_for_time": "2023-09-01T10:30:00Z",  
  "location": "New York",  
  "drone_shot": "High angle shot"  
}
```

- Same Operator can't create more than one booking for single time slot
- Operator can't book slot in past

This API also return the booking details

#Database Design

