🏢 What is VenueHub?

VenueHub is a website where:

- Businesses (like restaurants, hotels, party halls) can **list their spaces**.
- Users (like you and me) can **book those spaces** for events (weddings, parties, etc.).

🚀 What does it do?

🔐 User & Business Accounts

- Users sign up to **book venues**.
- Businesses sign up to **list venues**.
- Each has their own dashboard to manage things.

🏠 Venue Listings

- Businesses add details like:
    - Venue name, location
    - Capacity (how many people it can hold)
    - Price, images, and services (like food, decoration)

🔎 Smart Search

- Users can search venues based on:
    - Event type (wedding, party...)
    - Location
    - Price
    - Capacity
    - Date

🧮 Booking System

- Users can:

    - Choose a date
    - Fill in booking info
    - Get confirmation and email

🧾 Invoice & Payment

- After booking, users get a **PDF invoice**.
- Payment can be done via gateways like Stripe, Khalti (to be added).

🛠️ My Dashboard

- Users: See all your bookings, change/cancel them.
- Businesses: Manage venues and check upcoming bookings.

🔔 Notifications

- Get email/in-app messages when:
    - Booking is confirmed or changed
    - Events are near

🧰 Technologies Used

Backend:

- **Django (Python)** — brain of the website
- **PostgreSQL / SQLite** — to store data

Frontend:

- **Django Templates + Tailwind CSS** — for layout & design
- **JavaScript (Vanilla)** — for interaction

Extras:

- **Pillow** — for image upload
- **WeasyPrint / xhtml2pdf** — for invoice PDF

- **Django Anymail** — to send emails

💻 How to Run This Project (Step-by-Step)

📦 Before You Start (Install These)

- Python 3.x
- pip (comes with Python)
- Node.js + npm (for Tailwind CSS)
- Git
- PostgreSQL (or just SQLite for local test)

⚙️ Setup Steps

1. **Clone the Project**

```
Git pull origin <branch_name>
git clone https://github.com/yourusername/venuehub.git

cd venuehub
```

2. **Create a virtual environment**

```
bash
CopyEdit
python -m venv venv(name_of_virtual_environment)
source .\venv\scripts\activate  # (or source venv/bin/activate
 on Mac)
```

3. **Install required Python packages**

```
bash
CopyEdit
```

```
pip install -r requirements.txt
```

4. **Install Tailwind CSS**

```
bash
CopyEdit
npm install
npm run build
```

5. **Set Up Database**

```
bash
CopyEdit
python manage.py migrate
```

6. **Create Admin Account**

```
bash
CopyEdit
python manage.py createsuperuser
```

7. **Run the Server**

```
bash
CopyEdit
python manage.py runserver
```

Now go to http://127.0.0.1:8000

🔧 Git Commands Cheat Sheet

A simple and organized guide to essential Git commands for beginners and frequent use.

✨ Project Initialization

Initialize a Git repository

```
git init
```

📝 Remote Repository

Add a remote repository

```
git remote add origin <url_of_repo>
```

Change default branch name (e.g., from master to main)

```
git branch -M main
```

📝 Staging and Committing Changes

Add all files to staging area

```
git add .
```

Check current status

```
git status
```

Commit changes with a message

```
git commit -m "your_commit_message"
```

📡 Push and Pull

Push commits to remote repository

```
git push -u origin <branch_name>
```

Pull changes from a remote branch

```
git pull origin <branch_name>
```

📋 Branch Management

Create a new branch and switch to it

```
git checkout -b <branch_name>
```

Switch to another branch

```
git switch <branch_name>
```

Fetch all branch data from remote

```
git fetch
```

View all local branches

```
git branch -vv
```

View all branches (local and remote)

```
git branch -a
```

View only remote branches

```
git branch -r
```

⚠ Extra Useful Commands

Clone an existing repo to your local system

```
git clone <url_of_repo>
```

Remove a file from staging (if added by mistake)

```
git reset <file_name>
```

See commit history

```
git log
```

Undo last commit (keep changes in working directory)

```
git reset --soft HEAD~1
```

Check remote repository info

```
git remote -v
```

Delete a branch

```
git branch -d <branch_name>      # local
 git push origin --delete <branch_name>   # remote
```

Feel free to copy, save, and use this as your personal Git quick reference! 📂

📁 Project Structure (Basic Idea)

```
pgsql
CopyEdit
venuehub/
│
├── apps/
│   ├── users/      → login, register, profiles
│   ├── venues/     → venue info, images
│   ├── bookings/   → booking logic
│   ├── vendors/    → optional add-ons like photography
│   └── dashboard/  → user & business dashboards
│
├── templates/      → HTML files
├── static/         → images, JS, CSS
├── media/          → uploaded venue images
└── .env            → secret keys and settings
```

📌 Important Features

Feature	Description
🔐 Auth	Login/Signup for users & businesses
🏢 Add Venues	Businesses can add venue info
🔍 Search & Filters	Search by type, price, date, etc.
🗓️ Booking	Book and get confirmations
🧾 Invoices	Get booking PDFs
🎛️ Dashboards	Personalized view for user/business
🔔 Notifications	Email and app alerts
🛠️ Admin Panel	Full control using Django admin

☁️ Deployment Options

🌿 Heroku:

- Easy to deploy, free tier available

🐳 Docker:

- Use for advanced setup (optional)

🤙 For Beginners: Tips

- Work on **one feature at a time** (e.g., just user login first).
- Use **dummy data** to test (e.g., test venues).
- Don't worry about adding **all features** — start small and build slowly.
- **Ask for help** (StackOverflow, ChatGPT, or friends!).