

## FOLDER ARCHITECTURE -FRANCHISEE PERFORMANCE TRACKER

SaaS\_Project/

├── app.py

├── templates/

│ ├── base.html

│ ├── login.html

│ ├── register.html

│ ├── forgot\_password.html

│ ├── reset\_password.html

│ ├── admin\_dashboard.html

│ ├── franchisee\_dashboard.html

│ ├── booth\_health.html

│ ├── profile.html

│ ├── users.html

│ ├── sales\_entry\_form.html

│ ├── attendance\_log\_form.html

│ ├── ingredient\_reorder\_form.html

│ ├── admin\_sales\_page.html

│ └── report.html

└── static

│ └── images

│ └── VVLogo.png

## SQL ARCHITECTURE – FRANCHISEE PERFORMANCE TRACKER

### 1. Booth

- **Table Name:** booths
- **Description:** Represents a booth where sales and attendance are recorded.

Column Name	Data Type	Description
id	Integer	Primary key for the booth.
name	String(150)	Unique name of the booth.
location	String(255)	Location of the booth (optional).
users	Relationship	Relationship to the User model, representing users assigned to this booth.

### Routes Using Booth:

- /dashboard (Admin and Franchisee)
- /sales\_entry
- /attendance\_log
- /ingredient\_reorder
- /request\_booth\_change
- /approve\_booth\_change
- /reject\_booth\_change
- /booth\_health
- /get\_booth\_attendance/<int:booth\_id>

---

## 2. User

- **Table Name:** users
- **Description:** Represents a user of the application, either an admin or a franchisee.

Column Name	Data Type	Description
id	Integer	Primary key for the user.
username	String(150)	Unique username for the user.
email	String(150)	Unique email address of the user.
password_hash	String(256)	Hashed password for user authentication.
role	String(50)	Role of the user (admin or franchisee).
booth_id	Integer	Foreign key referencing the booths table (nullable).
approved	Boolean	Indicates if the user account is approved (default: false).
reset_token	String(6)	Token for password reset (nullable).
reset_token_expiration	DateTime	Expiration time for the reset token (nullable).
booth	Relationship	Relationship to the Booth model, representing the booth assigned to the user.

### Routes Using User:

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>• /login</li><li>• /register</li><li>• /approve_user</li><li>• /reject_user</li><li>• /forgot_password</li><li>• /reset_password</li><li>• /profile</li></ul> | <ul style="list-style-type: none"><li>• /logout</li><li>• /dashboard</li><li>• /users</li><li>• /delete_user</li></ul> |
|---|--|

---

### 3. SalesEntry

- **Table Name:** sales\_entries
- **Description:** Represents a record of sales made at a booth.

Column Name	Data Type	Description
id	Integer	Primary key for the sales entry.
user_id	Integer	Foreign key referencing the users table (nullable).
date	Date	Date of the sales entry.
drink_name	String(100)	Name of the drink sold.
quantity	Integer	Quantity of the drink sold.
booth_id	Integer	Foreign key referencing the booths table.
booth	Relationship	Relationship to the Booth model, representing the booth where the sale occurred.

#### Routes Using SalesEntry:

- /sales\_entry
- /dashboard
- /generate\_report
- /sales

---

## 4. AttendanceLog

- **Table Name:** attendance\_logs
- **Description:** Represents attendance records for users at booths.

Column Name	Data Type	Description
id	Integer	Primary key for the attendance log.
user_id	Integer	User ID of the person logging attendance (no foreign key constraint).
username	String(150)	Snapshot of the username at the time of logging.
date	Date	Date of attendance.
present	Boolean	Indicates if the user was present.
booth_id	Integer	Foreign key referencing the booths table.
booth	Relationship	Relationship to the Booth model.

### Routes Using AttendanceLog:

- /attendance\_log
- /generate\_report
- /booth\_health
- /get\_booth\_attendance/<int:booth\_id>

---

## 5. IngredientReorder

- **Table Name:** ingredient\_reorders
- **Description:** Represents requests for ingredient reorders.

Column Name	Data Type	Description
id	Integer	Primary key for the reorder request.
user_id	Integer	User ID of the person making the request (no foreign key constraint).
username	String(150)	Snapshot of the username when the request was made.
booth_id	Integer	Foreign key referencing the booths table.
date	Date	Date of the reorder request.
ingredient_name	String(100)	Name of the ingredient being requested.
quantity	Integer	Quantity of the ingredient requested.
status	String(50)	Status of the request (default: 'Pending').
booth	Relationship	Relationship to the Booth model.

### Routes Using IngredientReorder:

- /ingredient\_reorder
- /update\_ingredient\_status
- /delete\_ingredient\_request
- /generate\_report

---

## 6. BoothChangeRequest

- **Table Name:** booth\_change\_requests
- **Description:** Represents requests made by users to change their assigned booth.

Column Name	Data Type	Description
id	Integer	Primary key for the booth change request.
user_id	Integer	Foreign key referencing the users table.
requested_booth_id	Integer	Foreign key referencing the booths table.
status	String(50)	Status of the request (default: 'Pending').
created_at	DateTime	Timestamp of when the request was created.
user	Relationship	Relationship to the User model.
requested_booth	Relationship	Relationship to the Booth model.

### Routes Using BoothChangeRequest:

- /request\_booth\_change
- /approve\_booth\_change
- /reject\_booth\_change

---

## RELATIONSHIPS BETWEEN CLASSES

- **Booth to User:** One-to-Many
  - A booth can have multiple users assigned to it, but a user can only be assigned to one booth at a time.
- **Booth to SalesEntry:** One-to-Many
  - A booth can have multiple sales entries, but each sales entry is associated with one booth.
- **Booth to AttendanceLog:** One-to-Many
  - A booth can have multiple attendance logs, but each attendance log is associated with one booth.
- **Booth to IngredientReorder:** One-to-Many
  - A booth can have multiple ingredient reorder requests, but each reorder request is associated with one booth.
- **User to SalesEntry:** One-to-Many
  - A user can have multiple sales entries, but each sales entry is associated with one user.
- **User to AttendanceLog:** One-to-Many
  - A user can have multiple attendance logs, but each attendance log is associated with one user.
- **User to BoothChangeRequest:** One-to-Many
  - A user can make multiple booth change requests, but each request is associated with one user.
- **User to IngredientReorder:** One-to-Many
  - A user can make multiple ingredient reorder requests, but each request is associated with one user.