Author

Name: Yuvraj Sharma
Roll number: 21f3000180
Email: 21f3000180@ds.study.iitm.ac.in
About: I am a bachelor with half a decade of experience with web development, entrepreneurship, research and development and music.

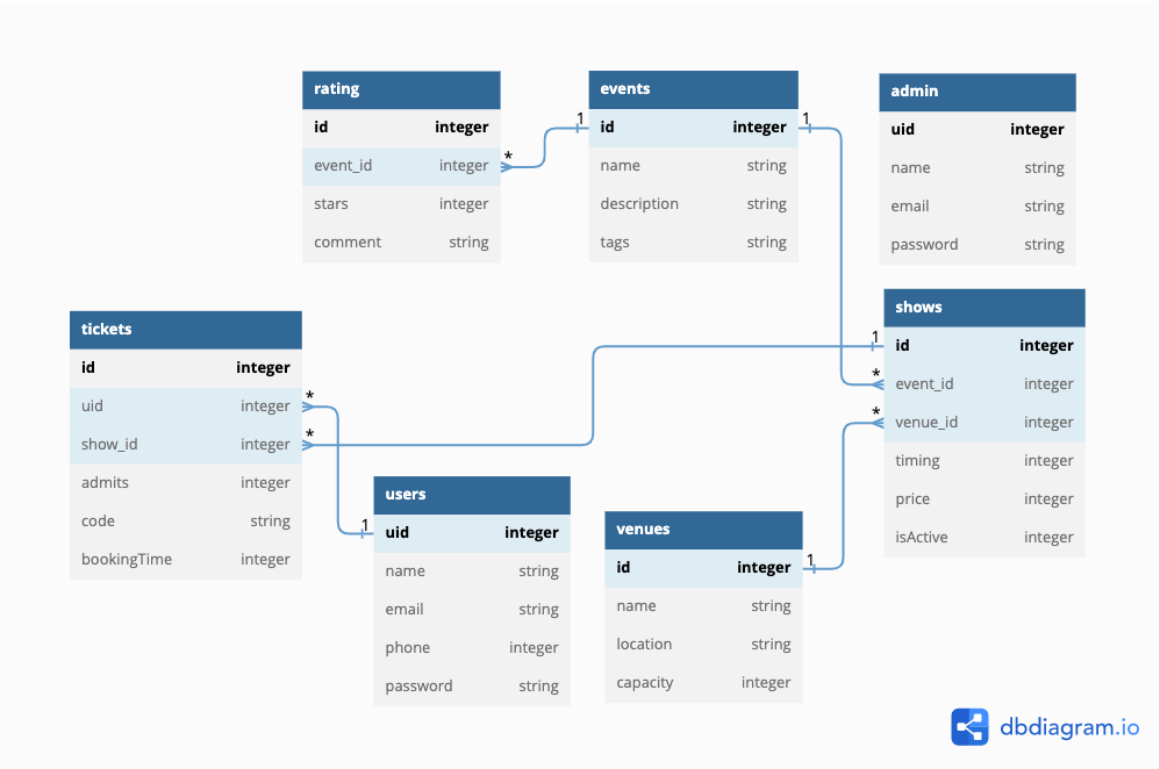
Description

The application is a show ticket booking platform, which allows searching for ongoing shows of a particular event or at a particular venue. The users can book tickets for various shows, the admin can create, update, delete and view the venues, shows, and events to be displayed to the users.

Technologies used

- ***Python*** (Used for creating the backend)
    - ***Flask*** (Used to create the web application)
    - ***Flask-SQLAlchemy*** (Used for interacting with the SQLite database)
    - ***Flask-RESTful*** (Used for creating REST APIs in Flask)
    - ***Jinja2*** (Used for templating in Flask)
    - ***Fernet*** (Used for encrypting / decrypting data within the application)
    - …and related dependencies for the above packages to function
- ***SQLite*** (Used for storing data)
- ***HTML5*** (Used to create a structure for the front end)
- ***CSS3*** (Used for custom styling)
- ***Bootstrap*** (Used for basic styling of the webpages)

Database Schema Design



The database was designed to scale, taking a vague inspiration from BookMyShow. There are 3 main entities: events, venues and shows. These are related to each other via one-to-many relationships, as shown in the diagram above. The `users` and `admin` table share similarities but have different levels of access within the backend.

API Design

The API has been created for the following features:

- **Rating**: To give ratings to a particular event and to retrieve the avg rating of a particular event.
- **Events**: To perform CRUD operations for events
- **Venues**: To perform CRUD operations for venues
- **Shows**: To perform CRUD operations for shows

The API is not secure, it currently allows unrestricted access to perform these operations.
The API documentation (api.yaml) file can be found in the `docs` folder in the project folder.

Architecture and Features

The project follows an iteration of the MVC model and follows the fundamental idea of separation of concerns. The models for each of the tables are defined in the `models` folder, the controllers which contain the functional methods with respect to the models are stored in the `controller` folder, the files which control the UI and the usage of controllers with authentication validations are stored in the `views` folder. Additionally, there are some utility methods / classes for authentication, database connection, configuration and exceptions, which are stored in the `utils` folder. All these folders are collectively part of the `app` folder. The SQLite database is stored in the folder called `db`. The folder named `docs` contains the API documentation file and database schema design. The `static` and `templates` folder contain the static assets, like CSS files needed for styling the appearance, and the design templates respectively.
Further details have been explained in the README.md file in the root folder.

Features implemented in the application are:

- Proper login and signup page for users and admin with symmetric encryption using Fernet
- Validation and authenticated access using a custom token-based authentication
- CRUD on venues and shows
- Booking and viewing tickets for users
- Searching for shows using various parameters like name, description and tags
- Displaying all the shows and their details to the user
- Giving reviews / ratings to a particular event

Video

https://drive.google.com/file/d/1LSzrVrwzMVazzN1dbjU5riJ2yw8ghYaL/view?usp=share_link