

## Report File

- There are 2 folders in my main project folder named frontend and web\_app.
- The web\_app folder contains all code related to backend of the website and the frontend folder contains all code relevant to frontend of the website.
- In the Web\_app folder there are 2 folders backend and instance, and main.py, requirements.txt file.
- The backend has all main files used to run the server properly.
  - Init.py : It contains all configuration related file.
  - Api.py : It contains all api information.
  - Auth.py : It has Login, Signup, Logout and JWT implementation.
  - Database.py : Database instance
  - Mail\_flask.py : Flask Mail instance
  - Mail.py : It has general mail implementation for sending mails.
  - Models.py : It has all models and tables information made in the database.
  - Views.py : It has all other classes implementation that is home, shows, venues etc.
  - Worker.py : It has all info regarding async jobs and celery tasks.
- The database file will be in the instance folder which will be inside Web\_app folder and our init file automatically creates our database file with a admin user if it does not exists.
- There is a templates folder inside Backend folder which has a report.html template which is used when we have to send monthly report to the user.
- Now my models has mainly 4 tables:
  1. User table:

It has id, email, password, name and isadmin as the attributes and it has a venues, bookings attributes to show relationship with the Venues and Booking tables. The default for isadmin is false for every user and it is true only for the admin it is adjusted by using the db browser only. I did not directly stored password in my database for security issues but stored it in the form of hash.
  2. Venue table:

It has id, venue\_name, place , capacity as attributes of a venue and owner, shows attributes are used for foreign key and establishing the relationship with the table shows.
  3. Show table:

It has id, show\_name, rating, start\_time, end\_time, tags, price, booked\_tickets as the attributes and hall, boo\_kings are the attributes used for foreign key and establishing relationship with the table bookings.
  4. Booking table:

It has id, tickets, booking\_date as the attributes and show, user are used to establish relationship with user and shows.
- Now my frontend folder has a src folder it has my most of the components required to start the vue server website.
- The src folder has following structure :
  - Assets : This folder contains image file for my website.
  - Router : It has a index.js file for my router setup.
  - Store : It has file to configure my store.
  - Views : It has all pages of my website.
  - App.vue : Used to connect all these things with the main.js file and start the server, can be considered as a main page in which all the other pages changes one by one like a tv show.

- The structure of my Views folder is given below:
  1. Book\_show.vue : Used to make new bookings.
  2. Bookings.vue : Used to show the bookings made by the user.
  3. Dashboard.vue : the place where the admin and user land after login or signup.
  4. Edit\_venue.vue : Used to edit the venue.
  5. HomeView.vue : General root page where the user will go after the site is started.
  6. Login.vue : Used to login admin and user.
  7. Show\_action.vue : Used to edit shows by the admin.
  8. Show\_add.vue : Used to add the shows by the admin.
  9. Signup.vue : Used to create new users.
  10. Venue\_add.vue : Used to add the venues.
- There is a requirements.txt file which specifies the libraries and header files used by this web application to run properly.
- There is a search box functionality for the user to search shows based on the name of the shows, genre of the shows and the ratings of the shows.
- The demonstration video link: [Video](#)