

## Report File

- There are two folder frontend and backend inside the MAD2 tickshow folder .
- The main.py file are present in frontend and backend respectively which will run the servers.
- There is a Webapp folder inside frontend folder which contains other important files like models, cache, celery\_worker, database, components for api and views and init file.
- The init file in my website has all configuration related code and the database initialization is also used to make the Webapp folder as a package so that main file can use it.
- The database file inside Webapp folder and our init file automatically creates our database file if it does not exists.
- There is a template folder inside Website folder which has jinja template .
- Now my models has mainly 4 tables:
  1. User table:

It has id, email, password, name and admin as the atributes and it has a venues, bookings atributes 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, name,location ,capacity as atributes of a Venue, venuesatributes are used for foreign key and establishing the relationship with the table show.
  3. Show table:

It has id, name, start time,end time, tag, rating, price, quantity as the atributes bookings are the atributes used for foreign key and establishing relationship with the table bookings.
  4. Userbook table:

It has id, category name, show name, quantity, total email as the atributes and show\_id are used to establish relationship with show table.
  - 5 Confirmbooking table:

It has id, category name, show name, quantity, total, overalltotal.
- The vuejs file has all the components required for authentication purpose that is:
  1. Login
  2. Logout
  3. Signup
  4. Createvenue
  5. Delete venue
  6. Edit venue
  7. Delete show
  8. Create show
  9. Edit show
  10. adminprof
  11. Home(usermain)

12. Bookings
13. Cart
14. Authnavbar
15. Usernavbar
16. Adminnav

- I used blueprint in my web app so that management of all Api become very easy and simple.
- There is a templates folder which has user\_report templates for send html to the user every month.
- There is a requirements.txt file which specifies the libraries and packages used by this web application to run properly.
- Jwt extended used for all the token based authentication
- Role Based authentication also specified for admin.
- All the three Asynchronous Job of daily reminder, monthly entertainment report and export csv have been implemented using redis , celery and MailHog for emails.
- I used Api for the CRUD operations on Show and Venue.
- There is a search box functionality for the user to search show by Veneu name, Show name, price .
- Booking page which has all the information of booked tickets by the user.

Video Link : [Click here](#)