PROJ4 – Security Audit

# **PART A**

1. **Check final application in a Desktop version of Chrome & Firefox + IOS and Android phones**

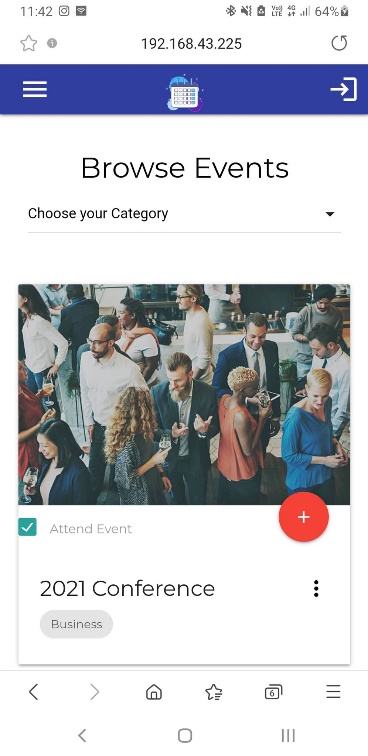
Final Applications are located at the below URLs:

Normal app: <https://malloriecini.com/>

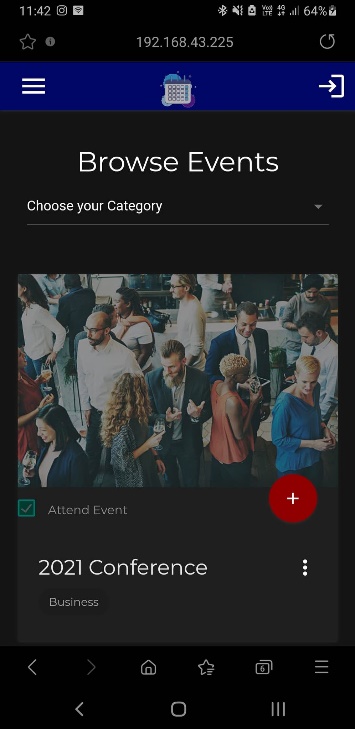
Admin app: <https://adminpanel.malloriecini.com/>

### Android

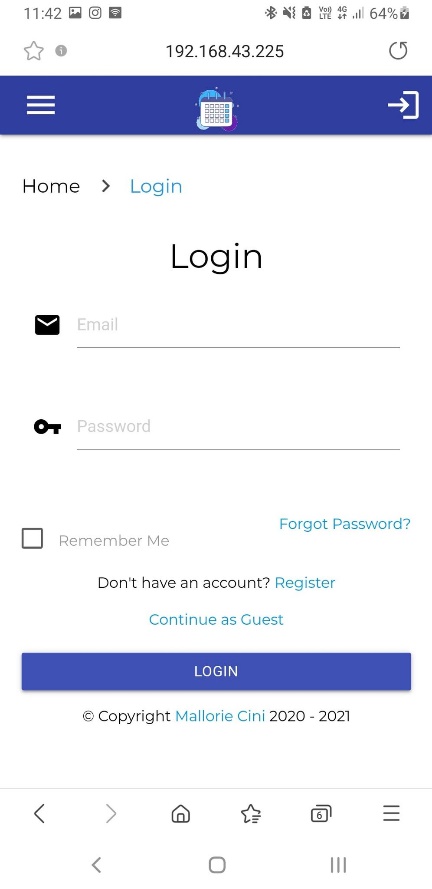
Home



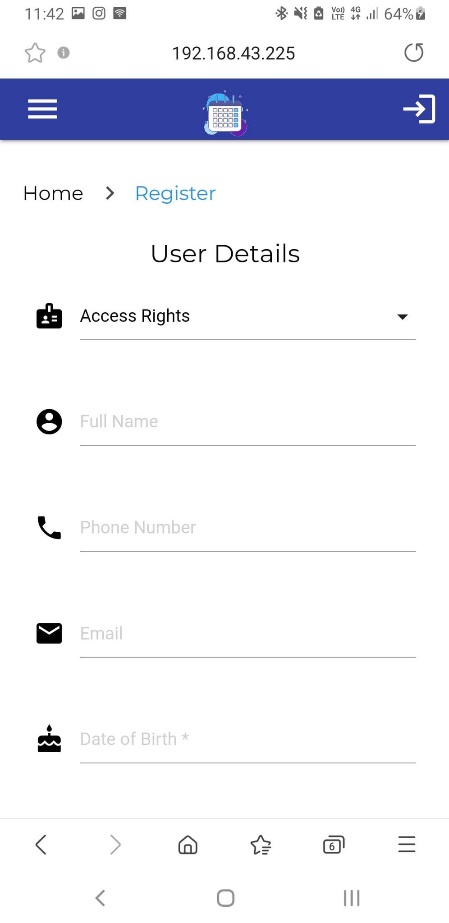
Dark Mode from system colours



Login

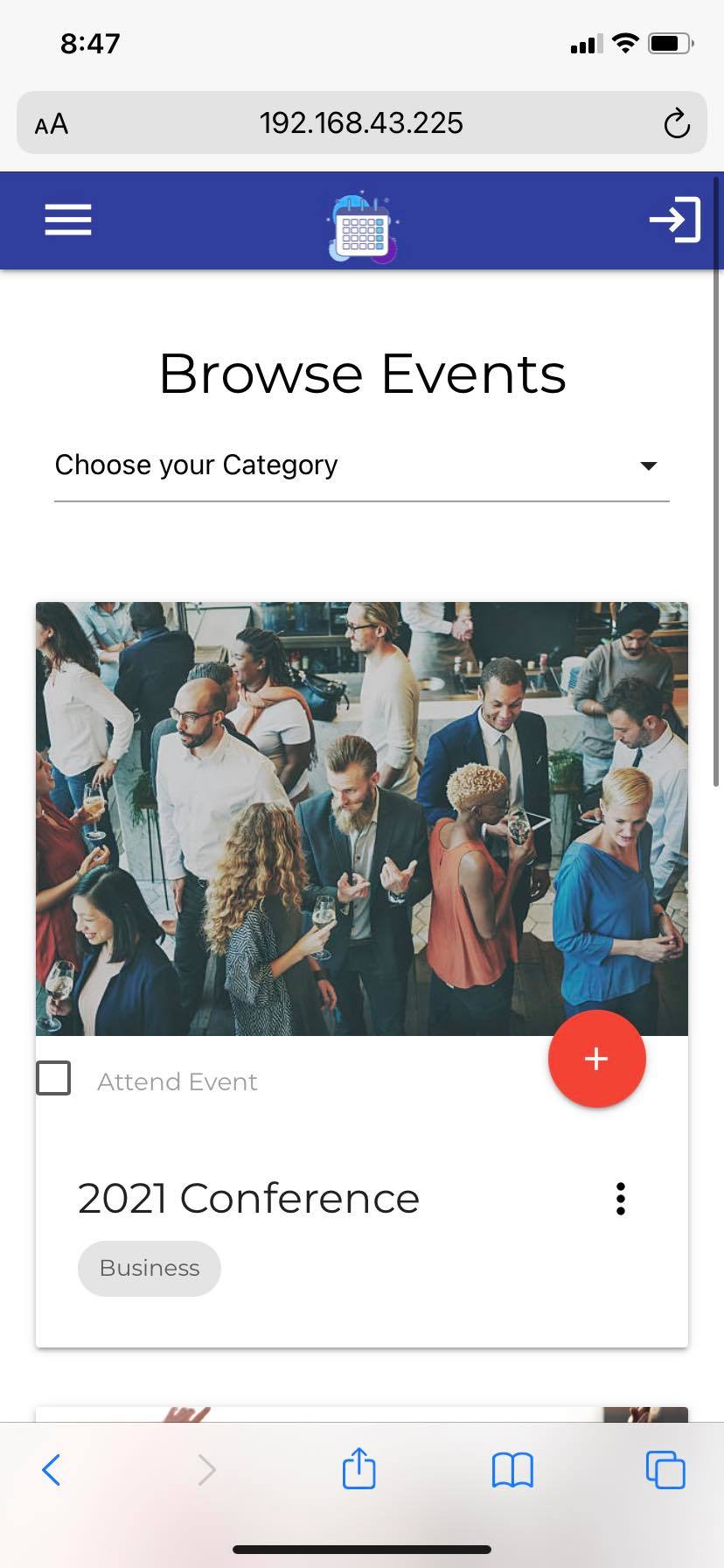


Register

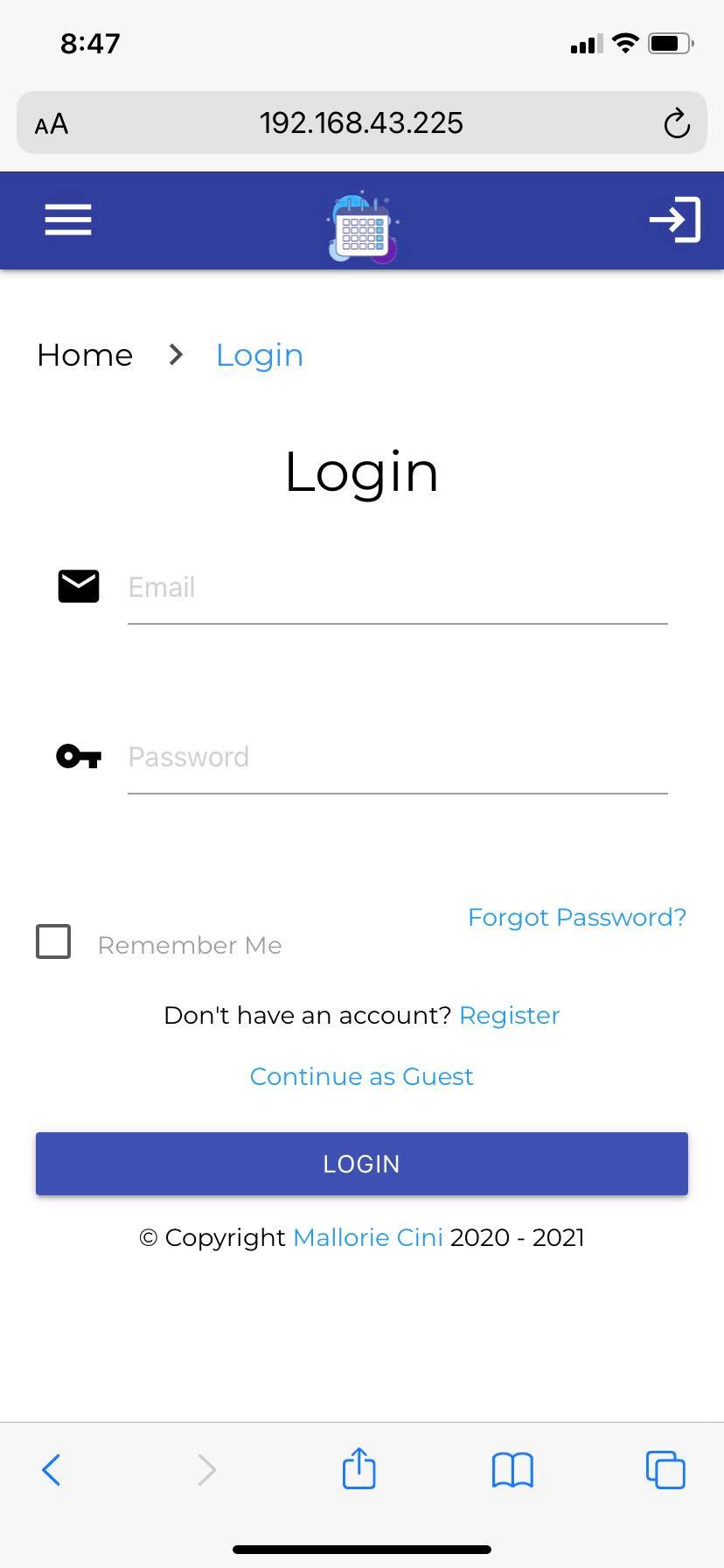


### iOS

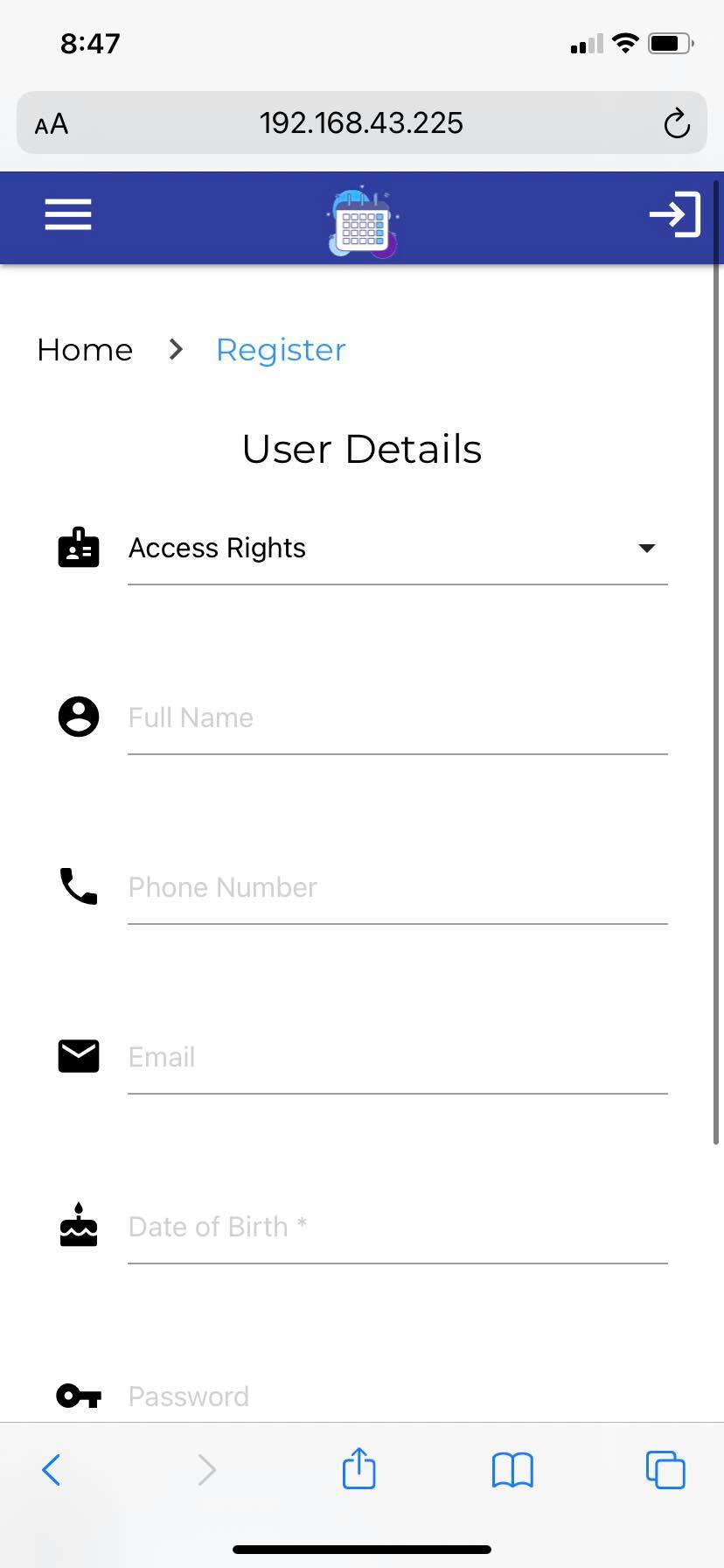
Home



Login



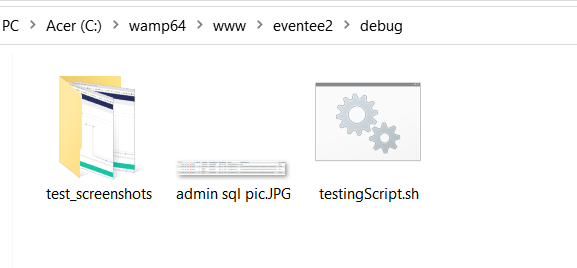
Register

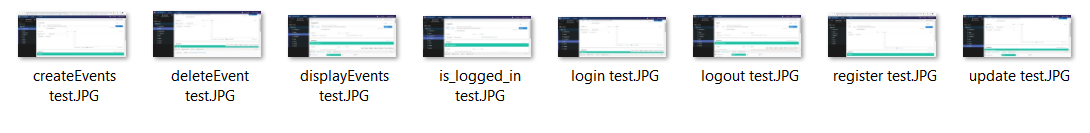


1. **Bring UX2, PROJ2 & PROJ3 together in one .zip, write a README describing installation for operations staff**

Github Link: <https://github.com/theavidcoder24/eventee2>

API **test screenshots** are located in the **debug folder** while the **test script is in the root**





1. **Include in the README all the technologies used in the app, places where they were used and versions you recommend**

The README.md is within the root of the directory which is also in the GitHub Repository.

<https://github.com/theavidcoder24/eventee2>

1. **Confirm functionality in relation to plan PROJ1 highlight areas that changed, or were not implemented.**

Completed and Functional

* Users can register successfully and is implemented
* Users can login successfully and is implemented
* Only users with access rights that are labelled “Admin” can login successfully to the admin panel which is implemented
* The user login status is able to be checked at all times which is implemented
* The user is able to logout thus ending the session and has been implemented
* Create events has been implemented into the application allowing the user to make custom events for the community to see
* User can display all events in a tabular view
* When a user brings up the update event form a function is implemented to autofill the form
* The user can update events by clicking a single button to open a modal and save the changes to the database
* The user is able to delete events from the database (will be implemented to be only the user specific events that the user has signed up to only removing it from their profile but keeping it in the events table database in the future)
* Facebook login api works only in deployment with a HTTPS certificate present but is implemented
* User is able to see events based on different categories on the home page
* The google maps api is present within the application but commented out so it can be activated if needed

Not Implemented as of yet

* User specific events added to their user id profile
* User will be limited to 5 active events that they can attend. For example, if a user has 5 active events that is happening in the future, the plus sign will be removed from browse events and will return when a user has 4 or less active events which will occur when the event is in the past.
* When user is signing up, they can select some category preferences and the app will recommend them when they are nearby said categories.
* See suggested events around the user’s current location in map view based on the user’s interested categories selected when registering.
* The application can pinpoint events that are near the user when they enter the app and click on “Explore Events” that also integrates with Facebook events giving the user multiple options.
* A past event that the user signed up to will appear grey.
* Users will be directed to the website of the event if they need to book tickets by having a link in the event description.
* To get your event featured at the top on Eventee you would have to pay a fee/subscription.
* Receive in-app notifications about upcoming events. If you are an event organizer you can receive notifications when someone has registered at your event.

1. **Write a roadmap section that confirms areas of development going forward. What bugs do you want to fix or functionality to extend?**

In future developments I will plan to make the application display notifications that will alert the user should they choose to set reminders about the events. The ability to upload a user profile image will also need to be implemented in the future updates of the application. A future implementation will let a user assign events to their user id so that they can see the events they want to attend.

Below is a boiler plate template that can be used for future implementations of said user specific events:


# **PART B**

1. **Secure app by resetting/removing admin passwords**User

Email: tracy@email.com

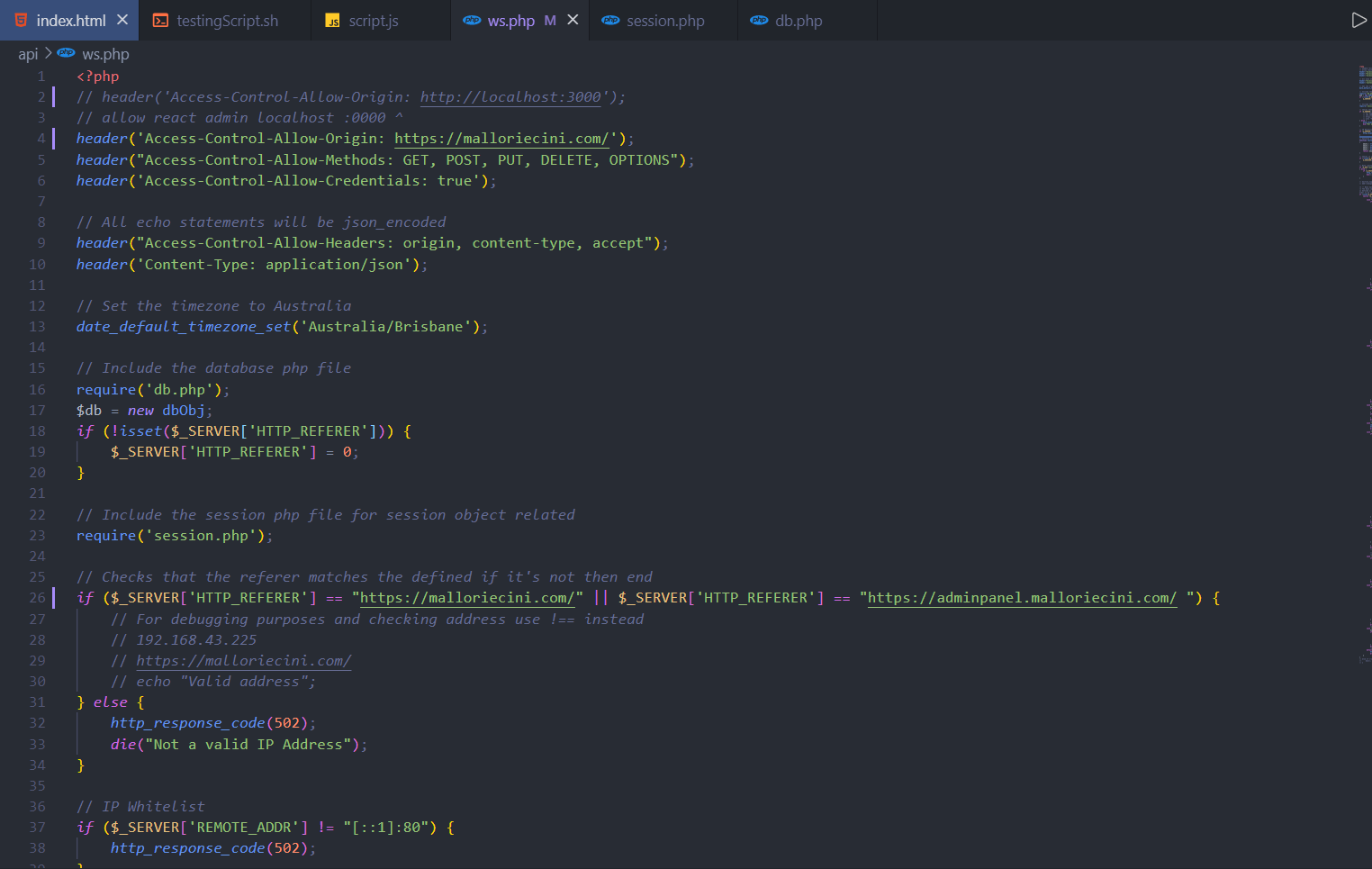
Password: User1

Admin

Email: derek@email.com

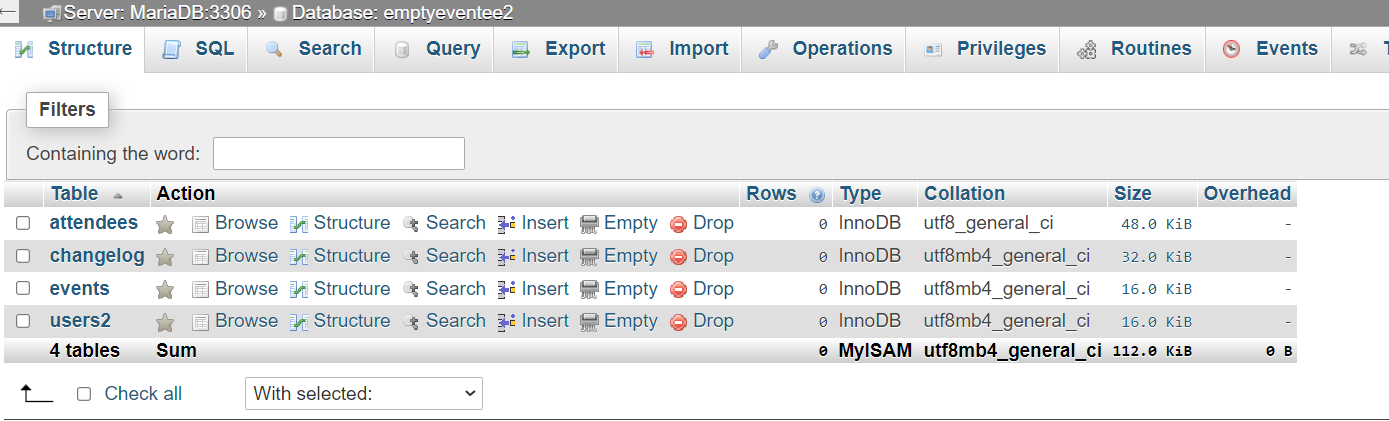
Password: admin123

1. **Remove localhost from referrer security, add hosting domain referrer**



1. **Remove log data & test data from app**

Empty SQL database located in api/sql named “emptyeventee2.sql”



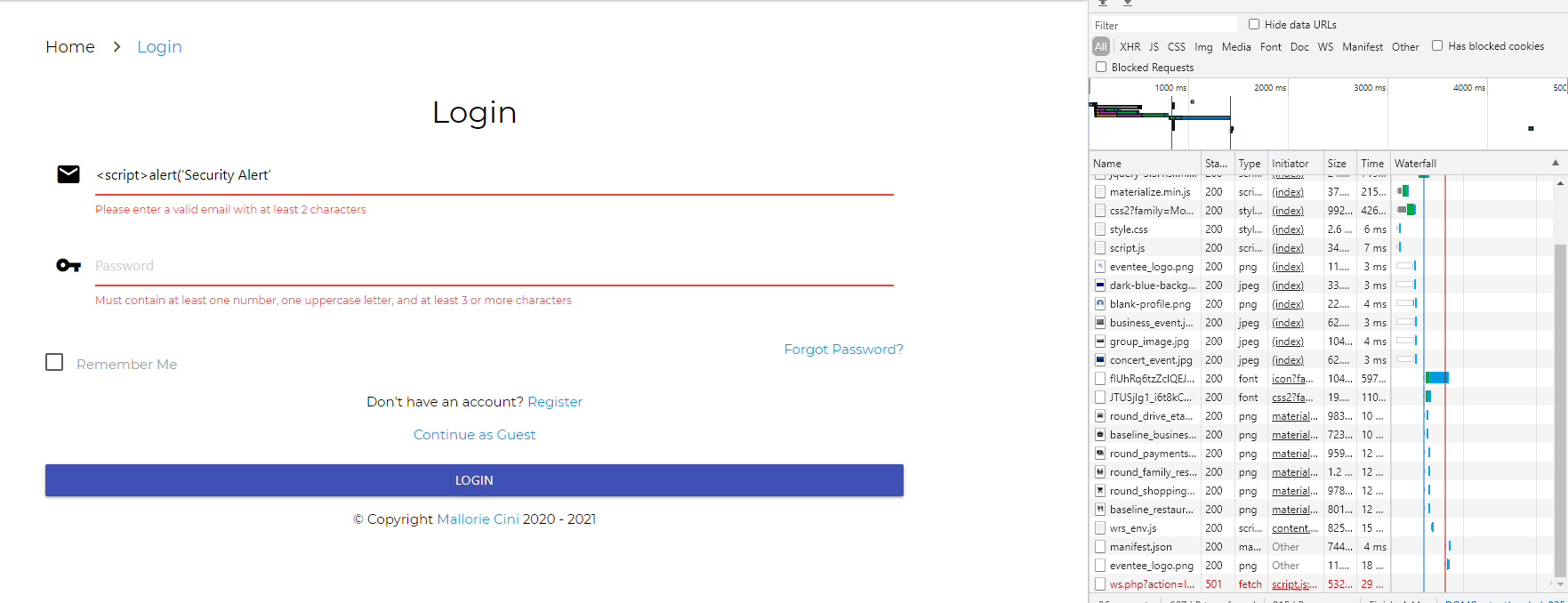
1. **Reset rate limits to once every .5 seconds & a maximum of 500 requests**

Option to switch back to 1000 requests for PROJ2 by uncommenting code + on line 53 added reset rate limits to once every .5 seconds

1. **Test with out of bounds data like negative numbers, Inject <script>alert(‘Security Alert’);</script> Update or delete on IDs that don’t exist**

Negative number:



Injection script alert:

Update on or Delete ID that doesn’t exist:

When the option to update or delete event appears there is no input field that the use can access to change the ID for update and as for delete method the user only has to click the one trash button to delete and it’s only on the same line as the event there’s no way to access the ID only in the console log does it show it after the update form is auto filled or a user deletes an event

1. **Screenshot results of 10 above and suggest a remedy**

Validation blocked the injection therefore no remedy was needed e.g., empty input fields)

