Final Submission - Application Final Component

Please ignore if Project File Name is Assignment 2, I've built up on previous assignment

Please use the provided credentials to Login or feel free to Sign Up with your own details.

Users for Login : v@v.com/test123 (3 Saved Connections); a@a.com/test123 (No saved Connections)

Note: You might not be able to login using the above the saved users due to BCRYPT, please create new User

Note: If you face any issues with npm install, it may be because I've used BCRYPT to salt hash passwords. It has few dependencies.

Ref: https://www.npmjs.com/package/bcrypt

Open PowerShell with administrator privileges and run this command, then proceed with bcrypt installation (For windows)

npm install --global --production windows-build-tools

Ref: https://github.com/kelektiv/node.bcrypt.js/wiki/Installation-Instructions

Login Page:

When user logs in, a database request is made to fetch users saved connection and added to session.

Refer: Controller/user.js (Line 96)

After logging in, clicking on your username on the top right in the navbar will take you to your saved connections

Along with server check, UI Check added to avoid null value submission.
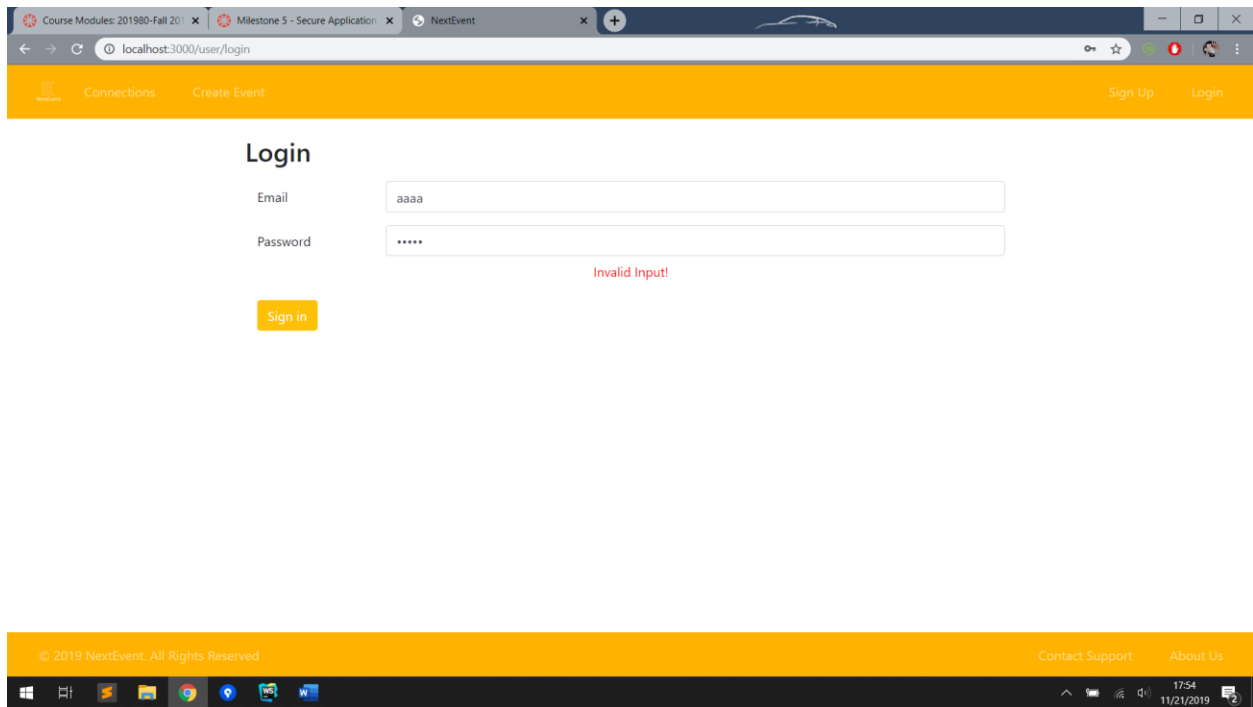
If the format of input is incorrect.
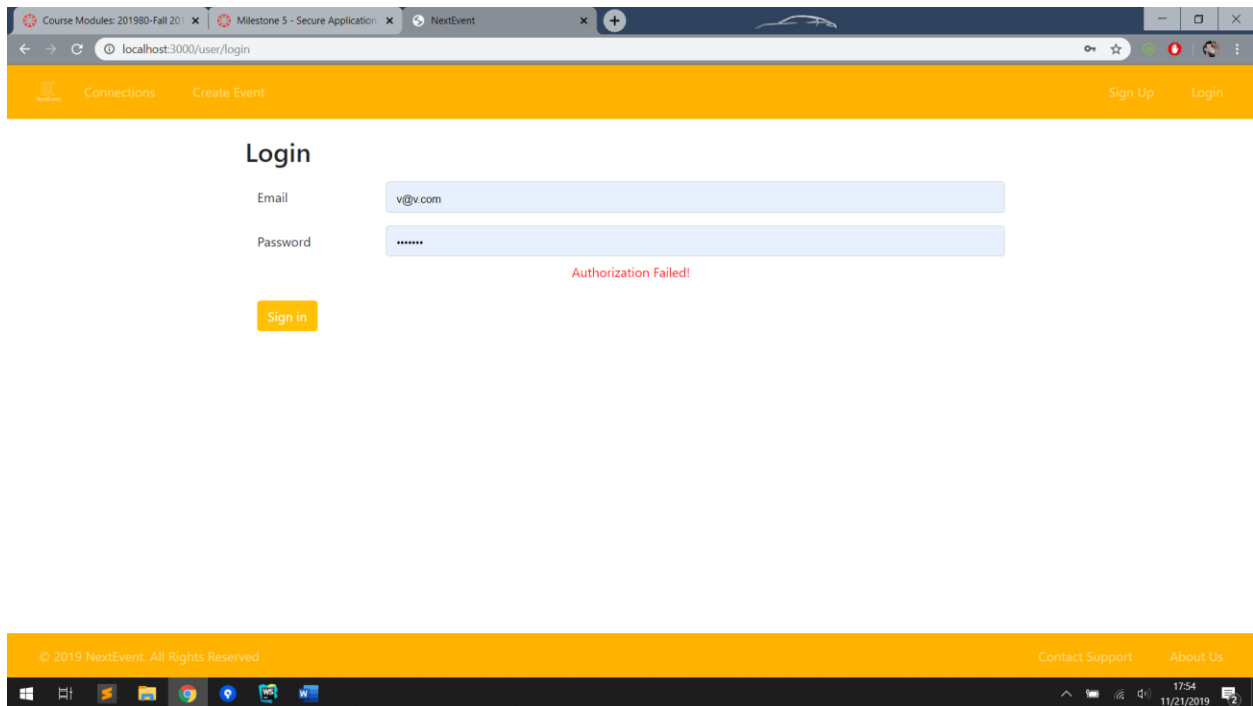
If incorrect password entered. Custom message provided.

Sign Up:

Like login, along with server check, UI Check added to avoid null value submission.

Email is validated for email format, first name and last name for alphabetic input only, and password validated for minimum length of 6.

When input data is incorrectly provided.
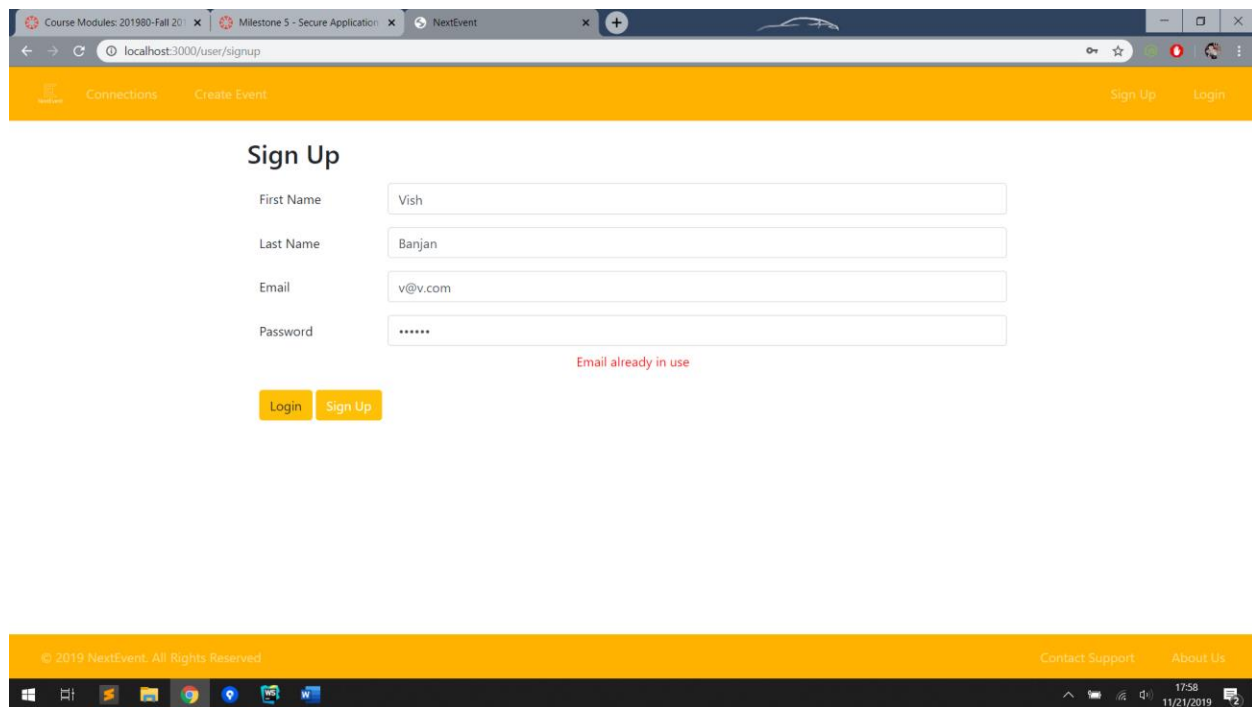
When trying to register using used email address. Custom message provided.

User Sign Up using BCRYPT :

```javascript
userSignUp: function (req, then, error) { //Sign Up , Create user object and store in database
    bcrypt.hash(req.body.inputPassword, saltRounds, cb: (err, hash) => {
        if (err) {
            error(err);
        } else {
            const user = new User({
                userId: mongoose.Types.ObjectId(),
                userFirstName: req.body.inputFirstName.trim(),
                userLastName: req.body.inputLastName.trim(),
                userPassword: hash,
                userEmailAddress: req.body.inputEmail.trim()
            });
            user.save().then(function () {
                then(user);
            }).catch(err => {
                error(err);
            });
        }
    });
});
```

User Login using BCRYPT :

```
userLogin: function (req, then, error) { //Check if email Exits
    User.findOne({
        userEmailAddress: req.body.inputEmail.trim()
    }).exec().then(doc => {
        bcrypt.compare(req.body.inputPassword, doc.userPassword, cb: (err, result) => {
            if (err) {
                error(err);
            } else if (result) {
                then(doc);
            } else {
                then(false)
            }
        });
    }).catch(err => {
        error(err);
    });
},
```
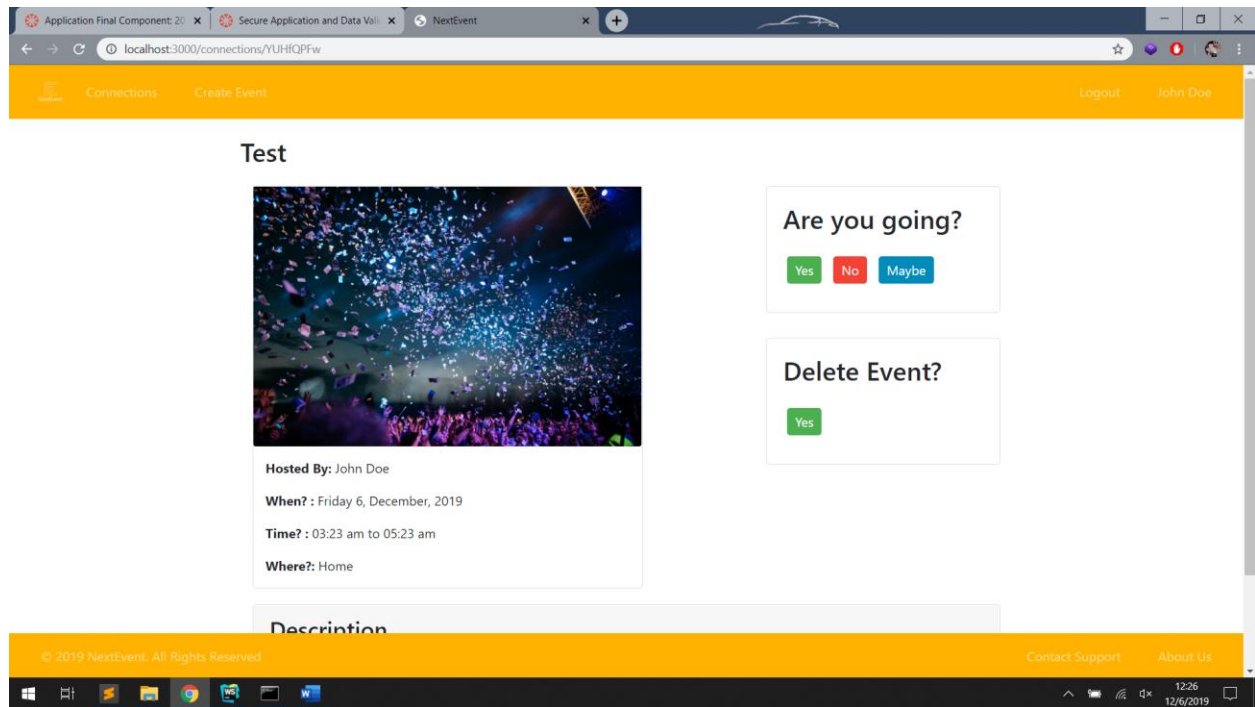
User Data in Database after using BCRYPT :
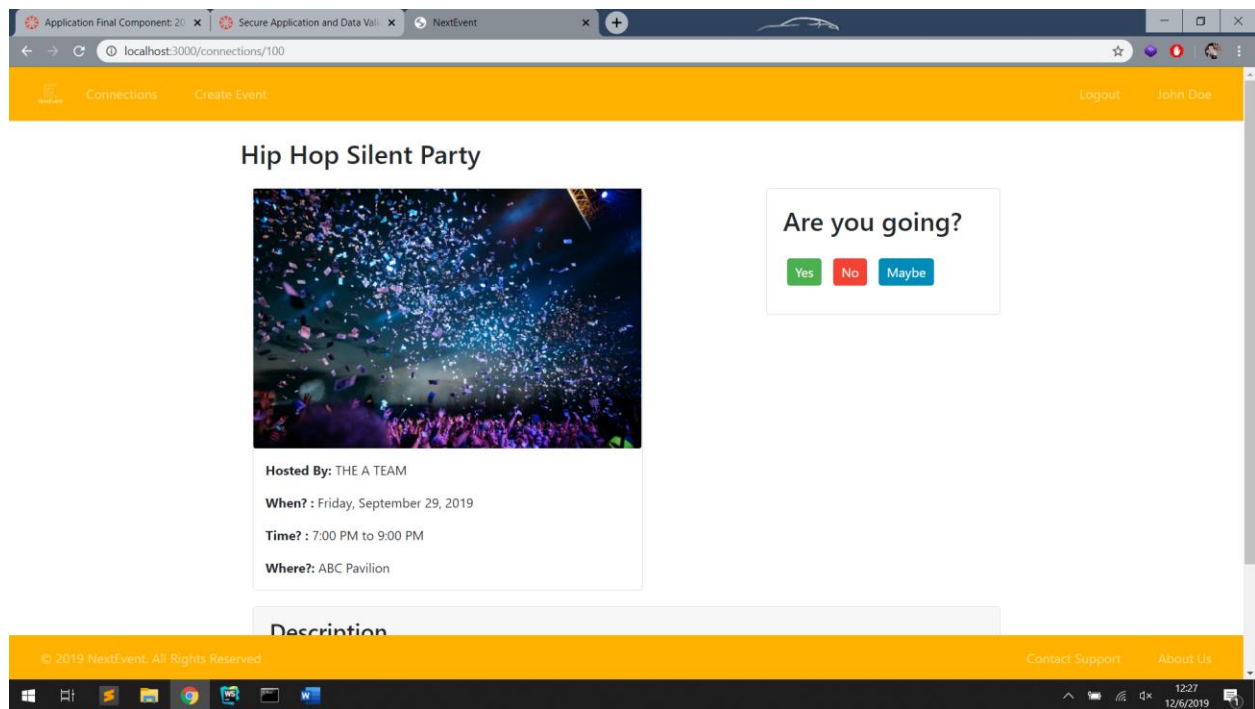
```
{
    "_id" : ObjectId("5dd95e4709393a1c58059078"),
    "userId" : "5dd95e4709393a1c58059077",
    "userFirstName" : "John",
    "userLastName" : "Doe",
    "userPassword" : "$2b$10$XZ9uD.WCxQLvMD77.p7Jj.m3t/Z.XB4w/n61M8I4DEV9PUDFsl9ku",
    "userEmailAddress" : "j@j.com",
    "__v" : 0
}
```

Delete Connections:

If the user logged in created particular connection , he will be shown a delete option.

If he's not the creator, He will not be shown the delete option. Once the connection has been deleted it won't be visible to other users even if they had given RSVP to it, it will be removed from their savedConnections.

Refer Client.js Line 74, for Event Listener code.

Refer Controller/Connection line 107, for delete connection code

Refer Controller/User line 28, Code for handling user having deleted connection in savedConnection

Securing Express HTTP Headers using Helmet :

Refer : App.js line 7