USER MANUAL

Course: ITIS 5166 - Network Based Application Development

Master of Science (Information Technology), University of North Carolina at Charlotte

Instructor name: Nadia Najjar

Prepared By: Akhil Reddy Yakkaluri

Application Name: Charlotte Connects

Charlotte Connects

Charlotte connects is event/meetup application which users use to meet and connect with others based on various events and categories. Various users can log into charlotte connects to sign up for various event across North Carolina based on their interests. Users can also use charlotte connect to post new events into the website which they are hosting.

The target user of this application are the people who like to meet new people, hangout, take part in activities outdoor. People who are like minded can come together and meet. Most of the people are free from their busy life on weekends, so they can chill out somewhere or take a hiking adventure etc.

The users can also host events from the application by logging as host users. This way others can come to his event and interact with each other. Charlotte Connects is a service used to organize online groups that host in-person events for people with similar interests.

Nowadays many people are so busy with their life they just go to work and stay home after work. People interacting with each other has become very less. People are becoming introverts. The application mainly aims to offers everyone an opportunity for anyone to gather for their chosen purpose-whether that be to explore passion, raise money, learn from others or protest.
The application helps you find and create events/meetups across Charlotte. Charlotte Connects is a platform for finding and building local communities. People use this platform to meet new people, learn new things, find support, get out of their comfort zones, and pursue their passions, together. At Charlotte Connects, you'll create and support a product that inspires millions of people to come together in

real life. The best part? You get to do it alongside some of the brightest, most passionate people across all disciplines and specialties.

Project Requirement:

- Separate css file is being used under Assets/stylesheets folder. It is accessed using Node Js's express.static library.
- The code is well documented at each function and routes for easy readability to someone new to this project.
- All the connections details, user details and user connections details are stored in mongo db under the name "nbad" into three different collections, "users","userprofiles","connections".
- The application is made secure by implemented sha512 hashing by including the salting and cross site scripting is prevented by validating the input values.
- The requirements and design document talk about the design process, various pages, database etc
- User manual talk about the flow required for a new user to interact with the application.

General Requirement:

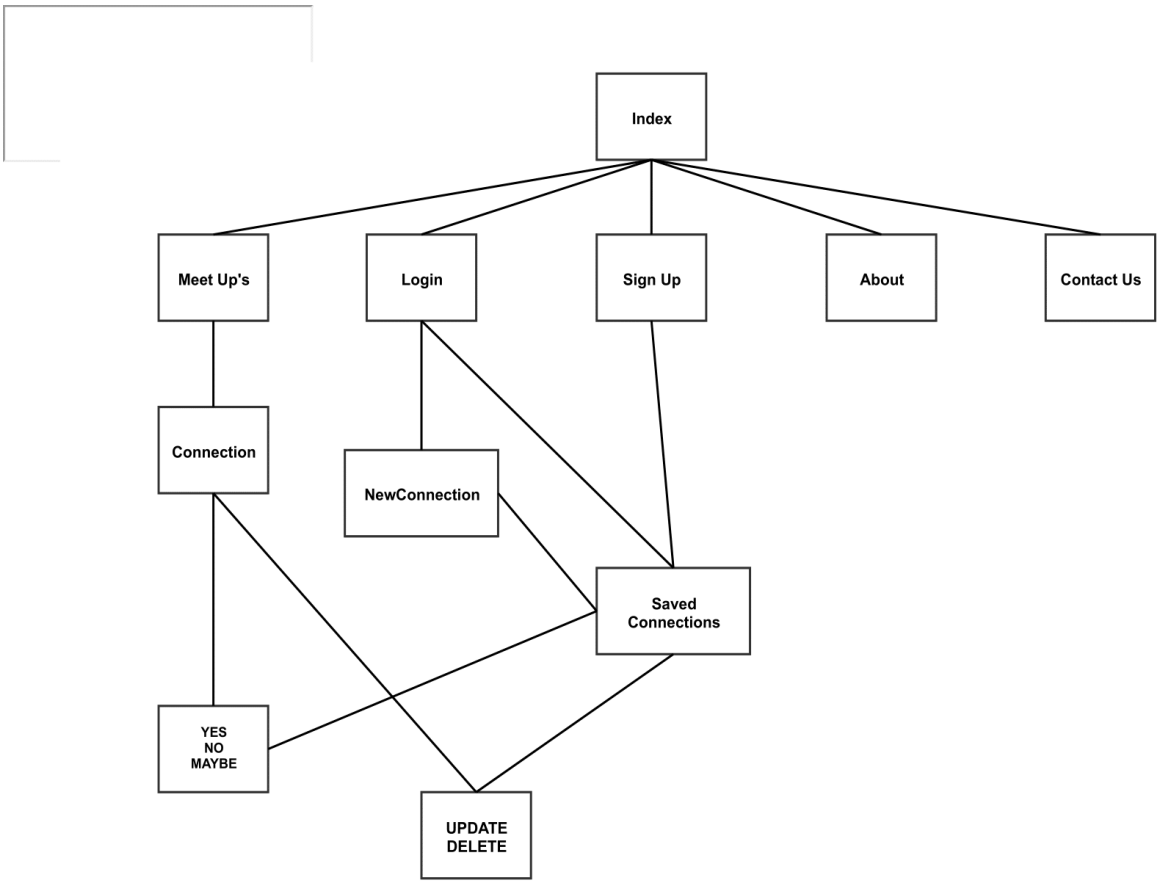
- Models: Four different javascript files are stored under model folder.
    - Connection.js: It specifies a class for connection having various.
    - User.js: It specifies a class for user having username, password etc.
    - Userprofile.js: It specifies class for userProfile having users connections and userID.
    - userConnection.js: It specifies class for userConnection having connection and rsvp.

- Views: Views folder consists of various ejs files for front end code.
- Controls: Controls are stored in routes. Two files are present: connectionController and profileController. These files consist of routes to handle various navigations in the application.

A registered user should be able to login into the application. The user should be able to RSVP for connections and update their RSVP or delete it. The user should be able to view all the connections in the application. The user should be able to create a new connection to host.

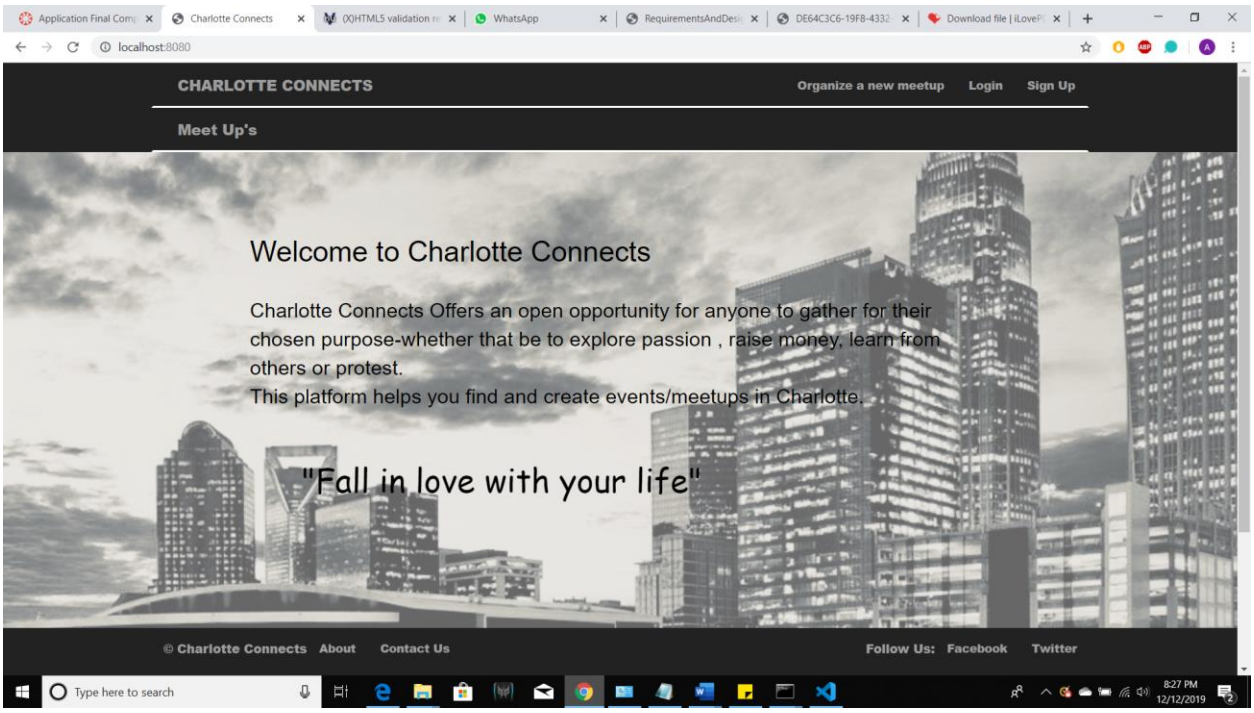
A general user can only view the connections and its details and can create an account in the application.

SITEMAP


Index
Meet Up's
Login
Sign Up
About
Contact Us
Connection
NewConnection
Saved Connections
YES
NO
MAYBE
UPDATE
DELETE

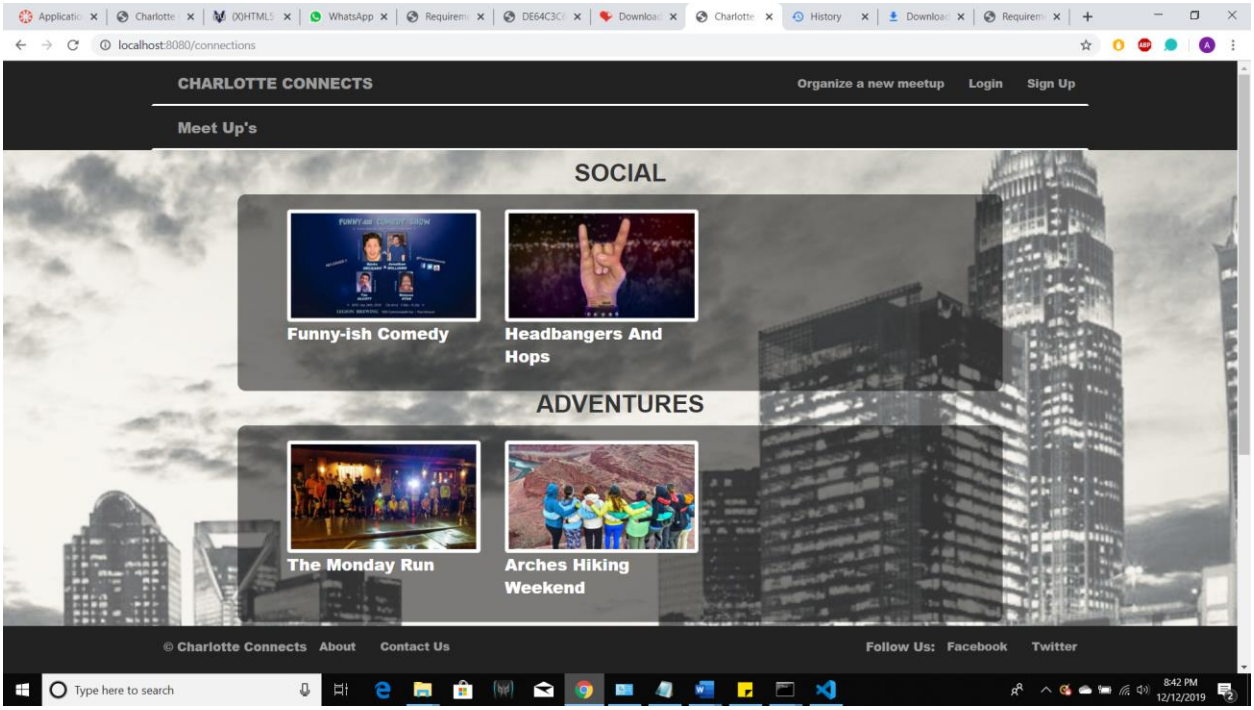
Page Design

Home



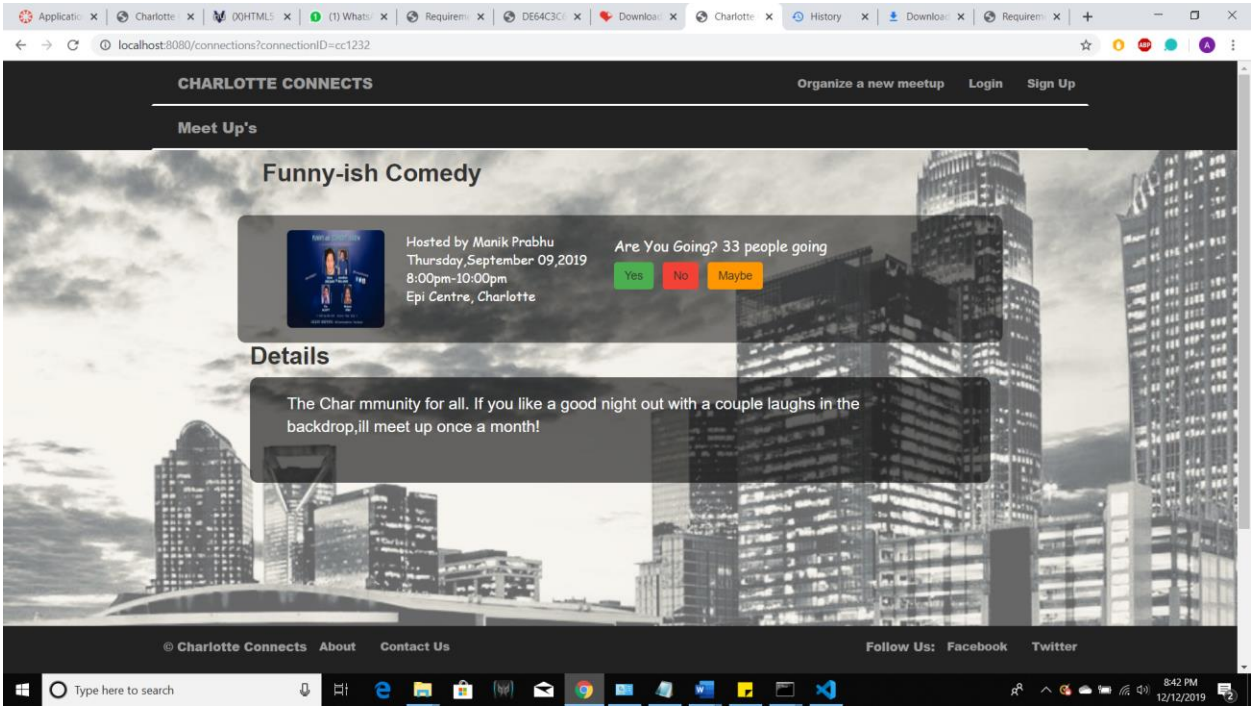
1. Name: index.ejs.
2. Purpose: This page is home page which is the start point for the application from where the user navigates to different pages.
3. Audience: Both the registered and unregistered users can access it.
4. Data: No logical data is present, only static application introduction is present.
5. Validations: No Validations.
6. Button or hyperlinks: Hyperlinks are present which wil take the to the appropriate pages.

CONNECTIONS



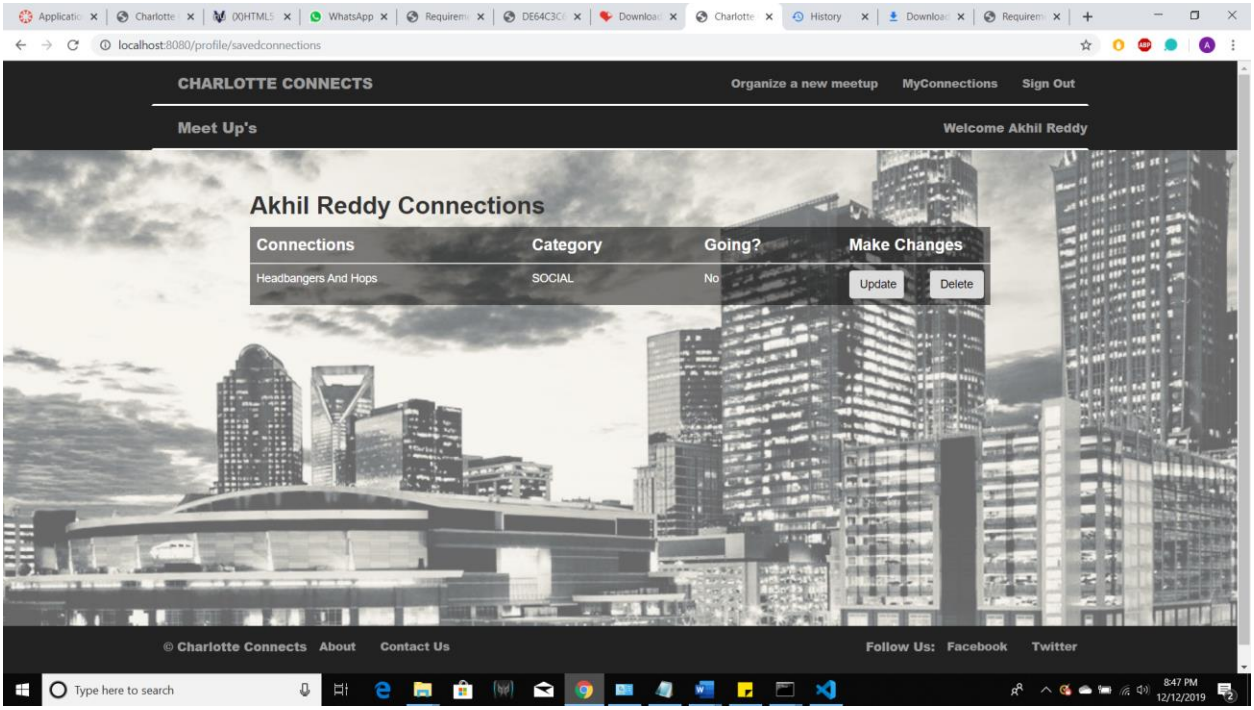
1. Name: connections.ejs.
2. Purpose: This page is connections page which shows all the connections available in the application.
3. Audience: Both the registered and unregistered users can access it.
4. Data: The connections are shown based on categories. These are retrieved from the database and shown dynamically.
5. Validations: Output encoding.
6. Button or hyperlinks: Hyperlinks on each connection present will take it to the connection details page i.e. connection.ejs

CONNECTION



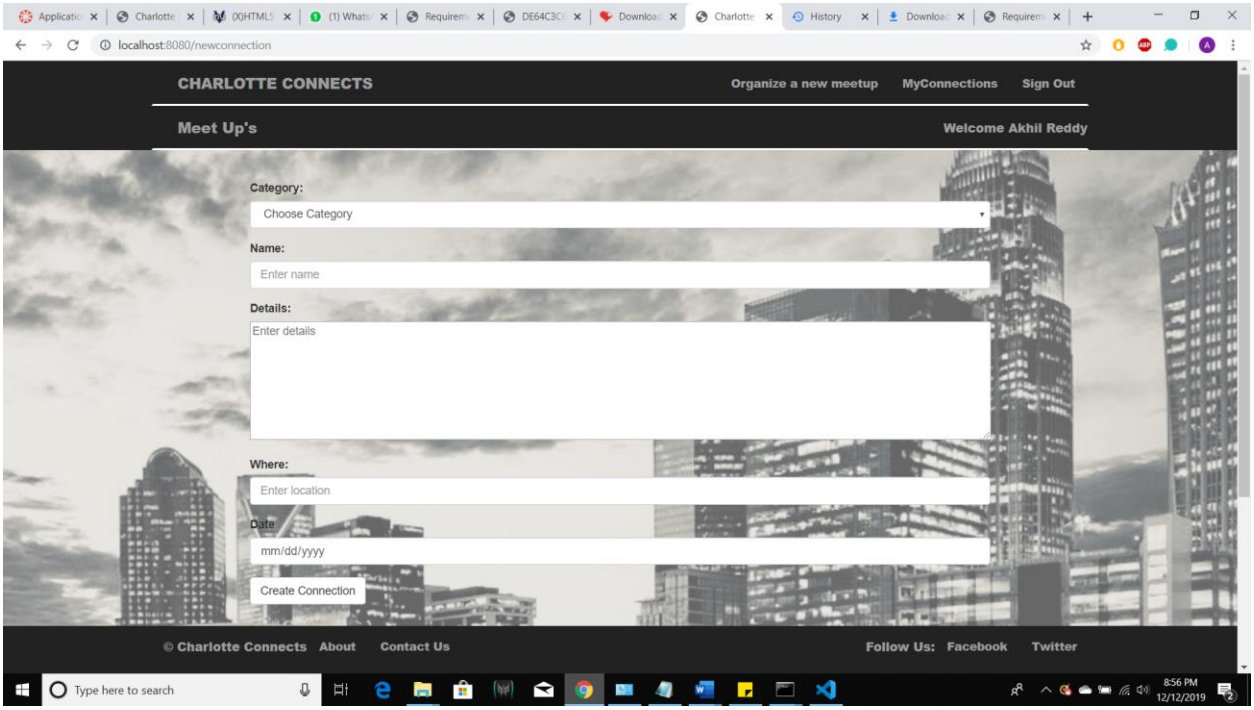
1. Name: connection.ejs.
2. Purpose: This page is connection page which shows the details of the selected connection. It also gives an option to RSVP the connection
3. Audience: Both the registered and unregistered users can access it.
4. Data: The connections are shown based on the selection. These are retrieved from the database and shown dynamically.
5. Validations: Output Encoding.
6. Button or hyperlinks: Hyperlinks on header, footer and navigation will take it the respective pages. The buttons will take it to the saved connections page i.e. myConnections.

MY CONNECTIONS



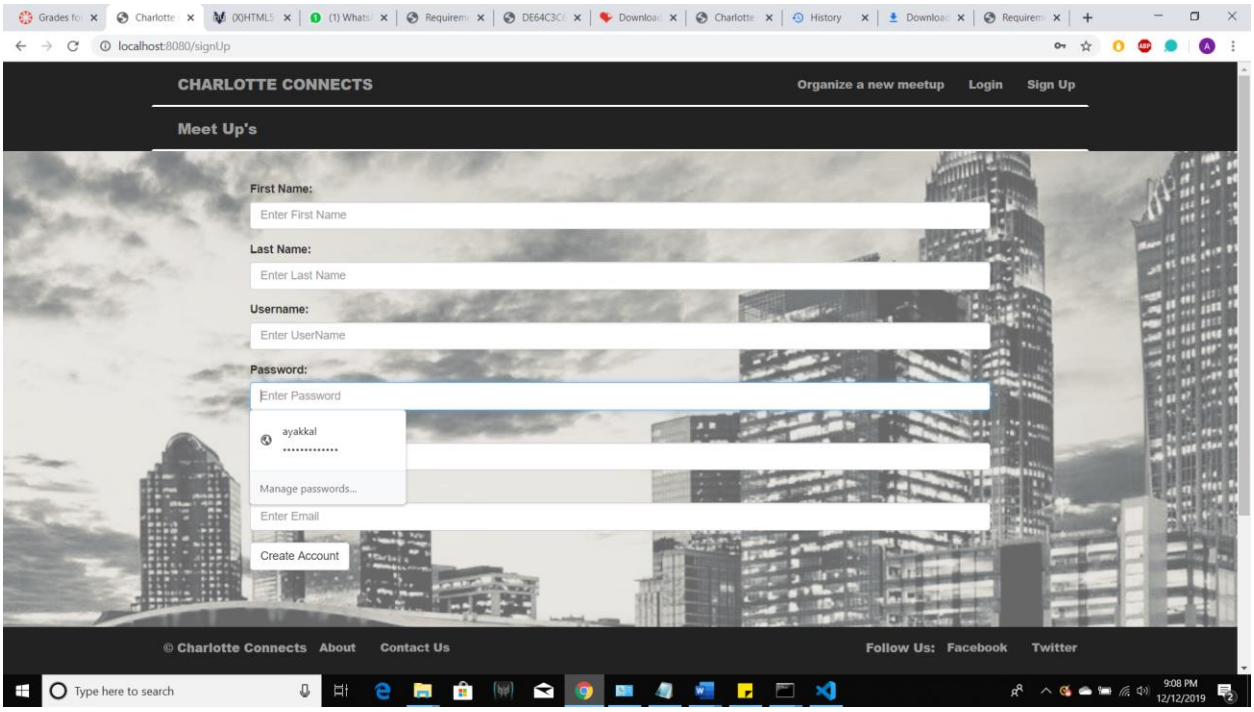
1. Name: savedconnections.ejs.
2. Purpose: This page is connection page which shows all the connections he RSVP'd to. He can also update his RSVP'd connection. It also gives an option to delete the connection from his list.
3. Audience: Only the registered user.
4. Data: The connections are shown based on the selection. These are retrieved from the database and shown dynamically. The logged in users name is displayed in the navigation bar.
5. Validations: Output Encoding.
6. Button or hyperlinks: Hyperlinks on header, footer and navigation will take it the respective pages. The buttons will take it to the saved connections page i.e. myConnections.

Organize New Connection



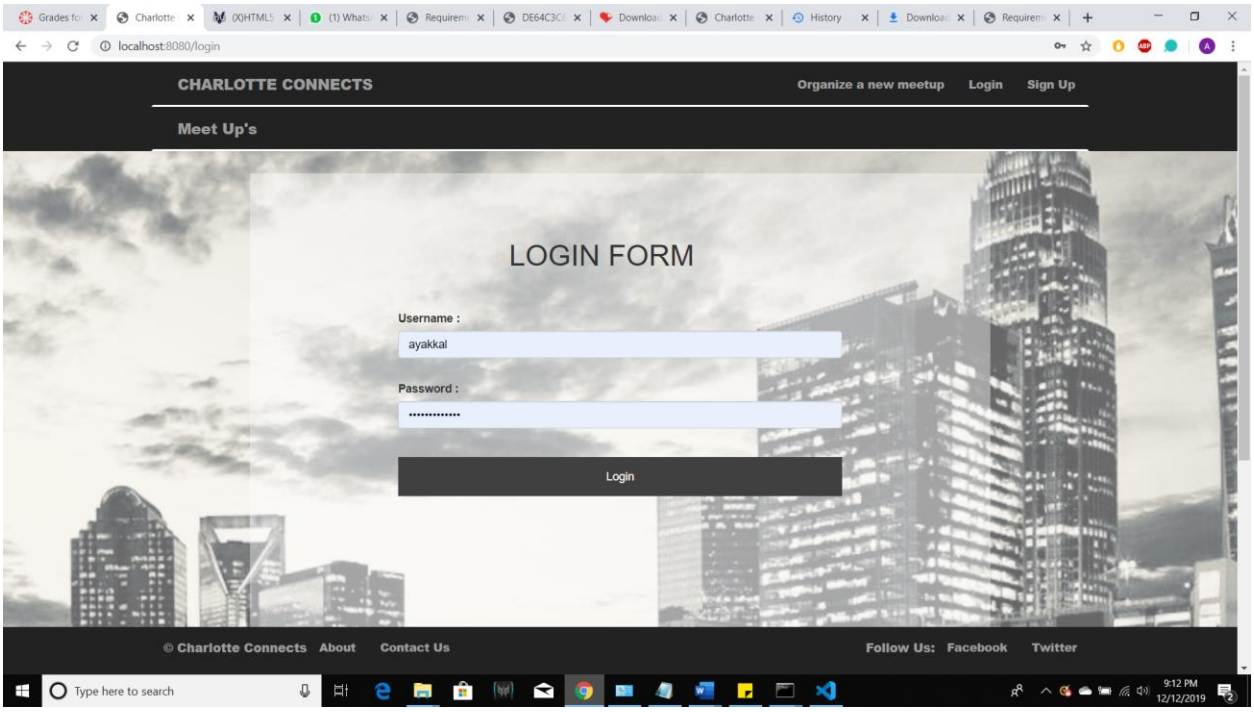
1. Name: newconnection.ejs.
2. Purpose: This page will allow the user to add new connections
3. Audience: Only the registered user.
4. Data: Input form is displayed. navigation bar.
5. Validations: Sanitization, empty fields checking and date valiation is done.
6. Button or hyperlinks: Hyperlinks on header, footer and navigation will take it the respective pages. The create button saves a connection to the connections page will take it to the connections page.

Sign Up



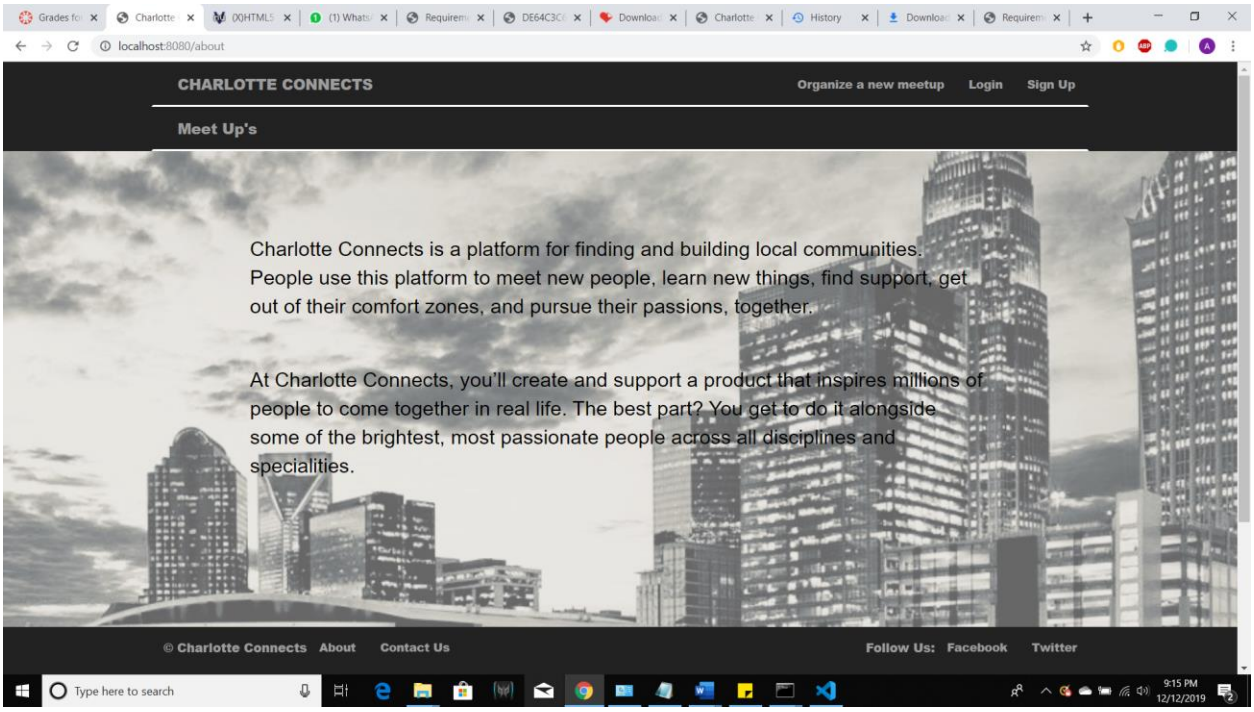
1. Name: signUp.ejs.
2. Purpose: This page will allow the user to register to the application
3. Audience: Only the new user.
4. Data: Input form is displayed.
5. Validations: Sanitization, empty fields checking, and date validation is done. Password is added with a salt and hashed automatically and stored.
6. Button or hyperlinks: Hyperlinks on header, footer and navigation will take it the respective pages. The create button saves the user details into the database, which will allow the user to directly login next time. This will take it to the saved Connections page.

LOGIN



1. Name: signUp.ejs.
2. Purpose: This page will allow the user to login to the application
3. Audience: Both the new user and registered users.
4. Data: Input form is displayed.
5. Validations: Sanitization, empty fields checking. Password matching is done.
6. Button or hyperlinks: Hyperlinks on header, footer and navigation will take it the respective pages. The login button will take the user to the saved connections page.

ABOUT



1. Name: about.ejs.
2. Purpose: This page will allow the user to access information about the application
3. Audience: Both the new user and registered users.
4. Data: Static data is displayed.
5. Validations: None
6. Button or hyperlinks: Hyperlinks on header, footer and navigation will take it the respective pages.

DATABASE STRUCTURE:

Database Name: “nbad”

Collections used

1. “users”: store the user data
    a. userID
    b. username
    c. password
    d. email
    e. firstName
    f. lastName
    g. salt
2. “userProfile”: stores user connections
    a. userID
    b. userConnections:{
    connectionID
    rsvp
    }
3. “connections”: stores all the connection details
    a. connectionID
    b. userID
    c. connectionTopic
    d. connectionDetails
    e. date
    f. time
    g. place
    h. connectionCategory
    i. imageUrl