Registration and Login:

Logged in users are able to create sessions and join sessions in order to plan out their hangout. If a user is not logged in when they click the Join Session button or Create Session button, they will be prompted to login.

Forms:

1. Signup form - allow users to create an account
2. Login form - allow users to login
3. Create Session form - allow users to create a session to plan out their hangout, creates a session code.
4. Join Session form - allows users to join a session to plan out their hangout, need a session code.
5. Itinerary Form - Users inputs all the aspects of their hangout like the name, activities, and the users email.

Blueprints:

1. Users
    a. / - (home)
    b. /signup - User creates account
    c. /login - User can login here
    d. /logout - Logs user out
    e. /dashboard - page displayed once the user has logged in
2. Session
    a. /session/create - Used for creating the session
    b. /session/join - used for joining the session
    c. /session/<session_id> - link displayed while you are in the specific session

Frontend Routes:

1. / – Home
2. /login – Login form
3. /signup – Signup form
4. /dashboard – Authenticated home
5. /session/create – Create session
6. /session/join – Join session
7. /hangout/:id – Hangout planning page

Tech Stack:

- Frontend: React
- Backend: Flask + Flask-Login + Flask-CORS
- Database: MongoDB (MongoEngine)
- Deployment: Render (https://hangouthub-koky.onrender.com)

Database (MongoDB):

In MongoDB we currently have the collections Users and Sessions to store user and session data. For users it will save their username, email, and password, while session will store the session name, id, start and end dates, a description, activities and participants.

Another Python Package or API:

We use **TicketMaster API** to show users events near their hangout location. This will give users an idea of potential activities they can do with their friends within the area.

We used the **Google Maps API** to embed a map showing the target hangout location so that users can view locations in the city that they want to hangout in.

We also used a React package called the **react-date-range** to get the calendar date range selection functionality.

NOTE: we ran into issues with the frontend and back end connection and were unable to get full client-server communication working in production, when we tried to deploy our app on Render. However, all functionality works as expected when run locally, and we have recorded a demo of that version