**APIs**

1. GET / : renders index.html for welcome page
2. GET /login : goes to google login, redirects to /login/callback, should not need to modify
3. GET /login/callback : gets authorization code google sent back, performs login at backend, redirects /newuser if user not found in database, else redirect to /view
4. GET /newuser : renders newuser.html, which contains forms for user to put information for creating user
5. POST /newuser : gets the forms from user and insert into user table, redirect to ?
6. GET /logout : logout

**Webpages**

1. Index (/) (A welcome homepage, with a button to /login for google login, A list of current streaming channel; After a member login, additional options (links) are shown to the user including: logout, start streaming, my profile)
2. Video player (/view/ID)
3. Streamer Dashboard (/stream) (after a member clicking on the “start streaming” button, he/she will be redirected to the streamer dashboard. Choose a related streaming type from the drop-down list and click on “build pipeline”, the backend program will start to build a personal streaming pipeline, while finish, the information the member needs to start streaming using OBS will be shown. Also, in this dashboard, the streamer is able to see his/her own live-streaming (the same as what other audiences can see) and other information)
4. Company Dashboard (/company)
5. Profile (/profile)

**Database**

1. User
   1. Id (from google)
   2. Name (from google)
   3. Email (from google)
   4. Profile pic (from google)
   5. UserType (streamer/viewer/company)
   6. ~~StreamCategory (if selected streamer)~~ this can be inputted while starting the streaming
2. WatchHistory
   1. Viewer (userid)
   2. Streamer (userid)
   3. Time (timestamp)
   4. StreamCategory
3. Streams
   1. Id
   2. Category
   3. Viewer

**Logic**

Every time a viewer views a stream, add a row to view table. To calculate the statistics, join the view table with the user table to get the user’s most viewed category.