ITIS 5166
Network-Based Application Development

Sessions and Cookies

Stateless HTTP

- HTTP is stateless protocol, i.e., a server does not keep track of information between each request/response cycle
- For example, a user logs in, then if the user browses to a different page (send a new request), the server does not remember the user
- Sessions and cookies are used to help the server and client maintain their transaction states
  - In the browser, state is stored in cookies
  - In the server, state is stored in sessions

Sessions and Cookies

- A session is an object that storing transaction states between a client and a server
  - user information, e.g., user id after a user is being authenticated
  - temporary states, e.g., flash messages
- Each session object has a unique id, called session id
- When a client sends its first request to the server, a session is created for this client
- The server then sends the session id to the browser, which is stored in a cookie
- The browser will include the cookie (session id) in the header of all subsequent requests

express-session module

○ express-session, a 3rd party module can be used to handle sessions

○ create a session middleware with the given options and mount the middleware on the application:

```
app.use(session({
   secret: //key for signing the session id cookie
   resave: //Boolean value
   saveUninitialized: //Boolean value
   cookie: //Object for setting session id cookie
}));
```

express-session documentation

How does it work?

- For each incoming request, the middleware checks if there is a cookie in the request header
- If there is, retrieve the session object matching the session id cookie, and attach the session object in the req object;
- otherwise, a new session object is created, set cookie of the res object
- To store or access session data, simply use the request property `req.session`

Session Store

- By default, sessions are stored in memory, which is not persistent
- The session data are cleared out every time we restart our app
- To have a persistent session store, we can store sessions in database
- express-session is compatible with a number of session stores

Demo: Sessions and Cookies