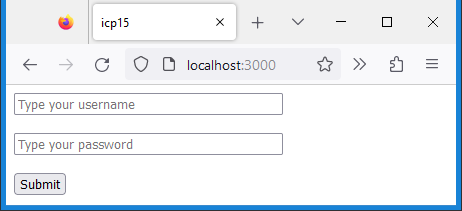
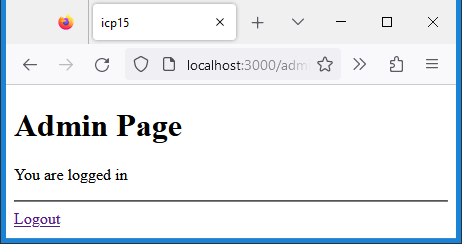
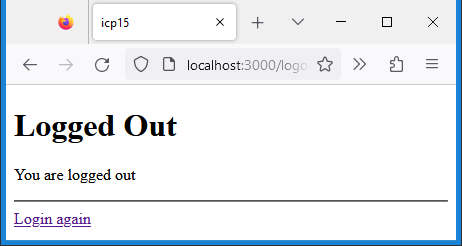
# In-Class Programming 15

Create the following JavaScript code to implement an express session



**Step 1:**

* Create JavaScript file icp15\_html.js to export object with 5 properties for html strings: top, bottom, login, admin, loggedOut:
  + top: Contains elements doctype to opening body tag, in head section character set utf-8 and title icp15
  + bottom: Contains closing body and html tags
  + login: Contains a form as shown, name attributes ‘user’ and ‘password’, button is a submit button, form action is ‘/login’, and method is post, two line break elements after text and password input, make inputs required
  + admin: As shown, logout anchor directs to ‘/logout’
  + loggedOut: As shown, login anchor directs to ‘/’

**Step 2:**

* Create JavaScript file icp15.js and create basic express web server (no need for 404 and 500 routes):
* Create constants for host and port
* Load express (variable app) and express-session (variable session), instantiate express
* Load file icp15\_html.js and store in variable html
* Create get route for root (‘/’), return login form wrapped in the html top and bottom values as shown in step 1
* Test the root path (login form), and view page source to make sure the html markup is correct

**Step 3:**

* Create a variable sSession at top level
* Use middleware session, set cookie secret, saveUninitialized and resave to true
* Use middleware urlencoded, setting property extended to true (needed for form data)
* In the get route for root, set variable sSession to the session object of the request
* Then test whether property user of sSession (from form name attribute user) exists:
  + If yes redirect (code 303) to route ‘/admin’ (res.redirect)
  + If no, return login form as in step 2
* Create get route ‘/admin’, again set variable sSession to session object of the request
* Test whether property user of sSession exists:
  + If no, redirect to ‘/’ using code 303 (res.redirect)
  + If yes, see part 4
* Test the application, do not click on the submit button yet (no implemented yet)
* First go to the root, you still should see login form
* Then type admin after the forward slash (localhost:3000/admin) in url address bar of your browser, it should redirect back to root since no user exists yet (have not submitted the form)

**Step 4:**

* Create a post route for ‘/login’:
  + Again, set variable sSession to session object of the request
  + Now the important part, assign the property user of the request body into sSession.user
  + Then redirect to ‘/admin’ with code 303 (res.redirect)
* In route “/admin”:
  + If sSession.user exist (continuation from part 3):
    - Send admin html wrapped in top and bottom properties back to client
* Create get route ‘/logout’:
  + Destroy session here, send loggedOut html wrapped in top and bottom html (no need for error object to keep it simple)

Test the application:

* Navigate to root (‘/’) 🡪 enter user and password 🡪 submit 🡪 logged in, admin page
* Remove admin from url (basically going back to root ‘/’) 🡪 navigating back to admin
* On admin page 🡪 logout 🡪 navigating to logout page 🡪 Click on login again 🡪/login
* Enter url /admin 🡪 since not logged in 🡪 redirected to login

Upload the following files to Canvas:

* Screenshots of the browser window showing login and admin pages (either pasted into a Word document or as an image)
* The JavaScript file icp15.txt (txt extension)
* The JavaScript file icp15\_html.txt (txt extension)

**Notes:**

* Upload the files individually (no zipping, compressing)
* Your code should contain some meaningful comments (if applicable)
* Your code should be well organized and formatted