# PSEUDO-CODE

Given: Templates for login, landing, logging out, profile manipulation, statistics, payments already existed and are styled, and all module prerequisites are met.

Map route for logging in

Login\_method():

Try to run this:

Sqlconnection = create connection

Create a cursor to traverse the data.

If the request method is a post method:

Query = a query to select data from the database from the provided

Username and password to check for a user.

Execute the query through the cursor.

If this user exists:

Load the users’ policies/homepage.

Else:

Load invalid user message.

Else if the request method is a get method:

Load the login page.

Load login page fallback

If you cannot run it:

Rollback all database updates to a previous safe state.

Map route for logging out.

Logout\_method():

Terminate session.

Load the user login page.

Map route for loading a policy.

Load\_Policy():

Try to run this:

Sqlconnection = create connection

Create a cursor to traverse the data.

If the request method is a post method:

Query = a query to select data from the database from the provided

Username and password to check for a user.

Execute the query through the cursor.

If this policy exists:

Load the policy.

Else if the request method is a get method:

Load the policy-page.

Load home page fallback

If you cannot run it:

Rollback all database updates to a previous safe state.

Map route for payments page

Load\_Payments():

Sqlconnection = create connection

Create a cursor to traverse the data.

If the request method is a post method:

Query = a query to select data from the database regarding the payments linked to the user and their policies

Execute the query through the cursor.

If the query has data:

Load the payments page.

If the user wants to make a payment:

Redirect to payment system.

If payment was successful:

Display successful payment message.

Reload the policy page.

Else:

Display payment error message.

Else:

Load error message

Else:

Load a message saying that the user has no payments yet.

Else:

Display error message.

Load home page fallback

If you cannot run it:

Rollback all database updates to a previous safe state.

Map route for statistics page

Load\_Statistics():

Sqlconnection = create connection

Create a cursor to traverse the data.

If the request method is a post method:

Query = a query to select data from the database based on their userid to fetch the appropriate data

Execute the query through the cursor.

If payment data exists:

Display the data and various statistics.

Else if the request method is a get method:

Load the statistics page.

Load home page fallback

If you cannot run it:

Rollback all database updates to a previous safe state.