HTML file: index.html

```
<!--
  CS351 Spring 2022
  Name: Bunla Kour
  Professor: Scott Spetka
 →


<!DOCTYPE html>
<html>

<head>
    <title>CS 351 HW7 Tan Ching Suen</title>
    <style>
        div {
            opacity: 0;
        }
    </style>
</head>

<body>
    <table nosave="" bgcolor="#ffff99" border="4" cellpadding="4" cellspacing="5" width="100%">
        <tbody>
            <tr>
                <td>
                    <table bgcolor="lightyellow">
                        <tbody>
                            <tr>
                                <td>
                                    <table border="4" cellspacing="5" cellpadding="4" cols="2" width="100%"
bgcolor="lightgray" nosave="">
                                        <tbody>
                                            <tr>
                                                <td>
                                                    <font color="#ff0808"><b>1)&nbsp;</b></font>
                                                    <b>Web Programming - Scott's Blog (ScottBlog)</b>
                                                    <p>
                                                        ScottBlog can be used to impart Scott's wisdom to the world!
His many "great thoughts" are posted into different categories, as described below. You will
implement the ScottBlog on this exam coding HTML/Javascript and Perl code to solve the
problem (see
                                                        <b>Implement ScottBlog Features</b>, below).
                                                    </p>
                                                    <p>
                            ScottBlog will keep track of users who have connected in the
order that they connect. In the initial implementation ScottBlog will just keep a log file, adding
the name of each user to the end of the log file each time anyone connects. Of course more
                            will be done with authentication in future versions of the
system.
                          </p>
                          <p>
                            In ScottBlog version 1 (ScottBlogV1) we will not worry about
synchronization issues that would arise if multiple users accessed ScottBlog concurrently. Of
course, multiple users may be using the system simultaneously, but not at exactly the same
instant.
                          </p>
                          <p>
                            You may structure the data that you will need to support
ScottBlog as simple file(s) in an organization of your choice.
                            <!--
    <table width=100% BORDER=4 CELLSPACING=5 CELLPADDING=4
BGCOLOR="#BBBBBB" NOSAVE >
    <tr><td align=center>
    <img src=files.jpg width=400>
    </td></tr>
    </table>
    -->
                          </p>
                        </td>
                      </tr>
                    </tbody>
                  </table>

                  <table width="100%" border="4" cellspacing="5" cellpadding="4"
bgcolor="#FFFF99" nosave="">
                    <tbody>
                      <tr>
                        <td>
                          <b>Implement ScottBlog Features</b>
                          <ul>
                            <li>
                              An initial "connect" form (shown in the diagram below)
asks for the user name and connects the user to the system.
                            </li>
                            <li>
                              Once connected, users will see a list of topics that they
can use to select which Scott Blog topics they want to see posts for. They may select any
number of r topics, including all of them.
```

```
                                        </li>
                                        <li>
                                           They will then see just the latest post that was posted to
ScottBlog for each of the selected topics (as shown in the diagram).
                                        </li>
                                        <li>
                                           The interface will also present a textarea input where
users can fill in ideas to post to ScottBlog (as shown in the diagram).
                                        </li>
                                        <li>
                                           Users can update the Scott Blog display. This allows them
to see if new posts came in since the display was last updated.
                                        </li>
                                        <li>
                                           Any changes in the list of topics of interest will be reflected
immediately in the display area when the user clicks the pick topics button.
                                        </li>
                                        <li>
                                           Any user can add a post to Scott Blog using the "Post
Idea" button in the "Post an Idea" section of the interface.
                                        </li>
                                        <li>
                                           The connected user will be displayed at the top of the
Scott Blog interface, along with the "Update Display" button (as shown in the diagram).
                                        </li>
                                     </ul>
                                  </td>
                               </tr>
                            </tbody>
                         </table>

                         <table width="100%" border="4" cellspacing="5" cellpadding="4"
bgcolor="#FFFF99" nosave="">
                            <tbody>
                               <tr>
                                  <td align="left">
                                     <table width="50%" border="4" cellspacing="5" cellpadding="4"
bgcolor="#FFFF99" nosave="">
                                        <tbody>
                                           <tr>
                                              <td colspan="4" align="center">
                                                 <b>Connect to Scott's Blogging System</b>
                                              </td>
                                           </tr>
```

```
                                                            <tr>
                                                                <td colspan="4" align="center">
                                                                    "Send Your Name to Start Blogging"
                                                                    <br /> Name:
                                                                    <input class="input--name" size="30" />
                                                                    <input type="submit" class="btn--name" />
                                                                </td>
                                                            </tr>
                                                        </tbody>
                                                    </table>
                                                </td>
                                            </tr>
                                        </tbody>
                                    </table>
                                    <div class="main">
                                        <table width="100%" border="4" cellspacing="5" cellpadding="4"
bgcolor="#FFFF99" nosave="">
                                            <tbody>
                                                <tr>
                                                    <td align="left">
                                                        <table col="4" width="50%" border="4" cellspacing="5"
cellpadding="4" bgcolor="#FFFF99" nosave="">
                                                            <tbody>
                                                                <tr>
                                                                    <td>
                                                                        <table col="4" width="100%" border="4"
cellspacing="5" cellpadding="4" bgcolor="lightgray" nosave="">
                                                                            <tbody class="blogSysTable">
                                                                                <tr>
                                                                                    <td colspan="4" align="left">
                                                                                        <b>Scott's Blogging System</b>
                                                                                    </td>
                                                                                </tr>
                                                                            </tbody>
                                                                        </table>
                                                                    </td>
                                                                </tr>

                                                                <tr>
                                                                    <td colspan="4" align="left">
                                                                        <b>Select Blogging Topics</b>
                                                                    </td>
                                                                </tr>
                                                                <tr>
```

```
                                              <td>
                                                <table col="4" width="100%" border="4"
cellspacing="5" cellpadding="4" bgcolor="lightblue" nosave="">
                                                  <tbody>
                                                    <tr>
                                                      <td colspan="3">
                                                        <table>
                                                          <tbody>
                                                            <tr>
                                                              <td>
                                                                <input type="checkbox"
name="testcase" value="Education" id="testcase1" /><label for="testcase1">education</label
                          >
                        </td>
                        <td>
                          <input
                            type="checkbox"
                            name="testcase"
                            value="Energy"
                            id="testcase2"
                            checked=""
                          /><label for="testcase2"
                            >energy</label
                          >
                        </td>
                        <td>
                          <input
                            type="checkbox"
                            name="testcase"
                            value="Nutrition"
                            id="testcase3"
                          /><label for="testcase3"
                            >nutrition</label
                          >
                        </td>
                        <td>
                          <input
                            type="checkbox"
                            name="testcase"
                            value="Hiking"
                            id="testcase4"
                            checked=""
                          /><label for="testcase4"
                            >hiking</label
```

```
                    >
                  </td>
                </tr>
              </tbody>
            </table>
          </td>
          <td>
            <input
              type="button"
              id="mybutton"
              onclick="topicDisplay()"
              value="Pick Topics"
            />
          </td>
        </tr>
      </tbody>
    </table>
  </td>
</tr>

<tr>
  <td colspan="4">
    <b>Post an Idea</b>
  </td>
</tr>
<tr>
  <td>
    <table
      col="4"
      width="100%"
      border="4"
      cellspacing="5"
      cellpadding="4"
      bgcolor="lightblue"
      nosave=""
    >
      <tbody>
        <tr>
          <td colspan="1">
            Pick a Topic
            <select name="topic" id="topic">
              <option value="Education">
                education
              </option>
```

```
                <option value="Energy">
                  energy
                </option>
                <option value="Nutrition">
                  nutrition
                </option>
                <option value="Hiking">
                  hiking
                </option>
              </select>
            </td>
            <td colspan="2">
              <textarea
                id="posttest"
                name="posttext"
                rows="4"
                cols="40"
                value="mypost"
              ></textarea>
            </td>

            <td colspan="1">
              <input
                type="button"
                id="mybutton"
                onclick="postIdea()"
                value="Post Idea"
              />
            </td>
          </tr>
        </tbody>
      </table>
    </td>
  </tr>

  <tr>
    <td colspan="4">
      <b>Scott Blog Display</b>
    </td>
  </tr>
  <tr>
    <td colspan="4">
      <table
        col="6"
```

```
                                    width="100%"
                                    border="4"
                                    cellspacing="5"
                                    cellpadding="4"
                                    bgcolor="lightblue"
                                    nosave=""
                                  >
                                    <tbody class="display">
                                      <tr>
                                        <td colspan="1">
                                          <b>Name</b>
                                        </td>
                                        <td colspan="1">
                                          <b>Topic</b>
                                        </td>
                                        <td colspan="2">
                                          <b>Post</b>
                                        </td>
                                      </tr>
                                    </tbody>
                                  </table>
                                </td>
                              </tr>
                            </tbody>
                          </table>
                        </td>
                      </tr>
                    </tbody>
                  </table>
                </div>

                <!--
    -->
              </td>
            </tr>
          </tbody>
        </table>
      </td>
    </tr>
  </tbody>
</table>
<script>
  const inputName = document.querySelector(".input--name");
  const btnName = document.querySelector(".btn--name");
```

```
      const blogSysTable = document.querySelector(".blogSysTable");
      const main = document.querySelector(".main");
      const display = document.querySelector(".display");

      function crobj() {
        let req;
        if (window.XMLHttpRequest) {
          req = new XMLHttpRequest();
        } else if (window.ActiveXObject) {
          req = new ActiveXObject("Microsoft.XMLHTTP");
        } else {
          alert("Failed to create request");
          req = "";
        }
        return req;
      }

      let globalName = "";

      const http = crobj();

      btnName.addEventListener("click", (e) => {
        e.preventDefault();
        main.style.opacity = 1;
        const row = blogSysTable.insertRow(-1);
        const c1 = row.insertCell(0);
        const c2 = row.insertCell(1);
        const name = inputName.value;
        globalName = name;
        c1.innerHTML = `<td colspan="3">
  <b>Connected User</b>: ${globalName}
</td>`;
        c2.innerHTML = `<td>
<input
  type="button"
  id="mybutton"
  onclick="updateDisplay()"
  value="Update Display"
/>
</td>`;
      });

      function updateDisplay(name) {
        http.open(
    "get",
    `/cgi-bin/hw7.cgi?myfunc=updateDisplay&name=${globalName}`
  );
  http.onreadystatechange = upadteDisplayRes;
  http.send(null);
}

function upadteDisplayRes() {
  if (http.readyState == 4 && http.status == 200) {
    const res = http.responseText;
    if (res) {
      let pres;
      pres = res.split(";");
          pres.pop();
      display.innerHTML = `<tr>
  <td colspan="1">
    <b>Name</b>
  </td>
  <td colspan="1">
    <b>Topic</b>
  </td>
  <td colspan="2">
    <b>Post</b>
  </td>
</tr>`;
      pres.forEach((e) => {
        const pe = e.split(",");
        const row = display.insertRow(-1);
        row.innerHTML = `<tr>
      <td colspan="1">${pe[0]}</td>
      <td colspan="1">${pe[1]}</td>
      <td colspan="4">${pe[2]}</td>
    </tr>`;
      });
    }
  }
}

function topicDisplay() {
  const selectedtopic = [];
  const T1 = document.querySelector("#testcase1");
  const T2 = document.querySelector("#testcase2");
  const T3 = document.querySelector("#testcase3");
  const T4 = document.querySelector("#testcase4");
  if (T1.checked) selectedtopic.push(T1.value);
  if (T2.checked) selectedtopic.push(T2.value);
  if (T3.checked) selectedtopic.push(T3.value);
  if (T4.checked) selectedtopic.push(T4.value);
  let requsetString = '';
  selectedtopic.forEach((e) => {
    requsetString += e += ',';
  });
  http.open(
    "get",
    `/cgi-bin/hw7.cgi?myfunc=topicDisplay&topics=${requsetString}`
  );
  http.onreadystatechange = topicDisplayRes;
  http.send(null);
}

function topicDisplayRes() {
  if (http.readyState == 4 && http.status == 200) {
    const res = http.responseText;
    if (res) {
      let pres=res.split("\n");
      display.innerHTML = `<tr>
    <td colspan="1">
      <b>Name</b>
    </td>
    <td colspan="1">
      <b>Topic</b>
    </td>
    <td colspan="2">
      <b>Post</b>
    </td>
  </tr>`;
      pres.forEach((e) => {
        const pe = e.split(",");
        const row = display.insertRow(-1);
            pe[0]=pe[0].replace(";","");
            pe[2]=pe[2].replace(";","");
        row.innerHTML = `<tr>
        <td colspan="1">${pe[0]}</td>
        <td colspan="1">${pe[1]}</td>
        <td colspan="4">${pe[2]}</td>
      </tr>`;
      });
    }
```

```
        }
      }

      function postIdea() {
        const row = display.insertRow(-1);
        const name = globalName;
        const topic = document.querySelector("#topic").value;
        const stuff = document.querySelector("#posttest").value;
        row.innerHTML = `<tr>
              <td colspan="1">${name}</td>
              <td colspan="1">${topic}</td>
              <td colspan="4">${stuff}</td>
            </tr>`;
        http.open(
          "get",
          `/cgi-bin/hw7.cgi?myfunc=postIdea&name=${name}&topics=${topic}&stuff=${stuff}`
        );
        http.onreadystatechange = postDisplayRes;
        http.send(null);
      }

      function postDisplayRes() {
        if (http.readyState == 4 && http.status == 200) {
          const res = http.responseText;
          if (res) {
            console.log(1);
          }
        }
      }
    </script>
  </body>
</html>
```

CGI file: pick.cgi

```
#!/usr/bin/perl
```

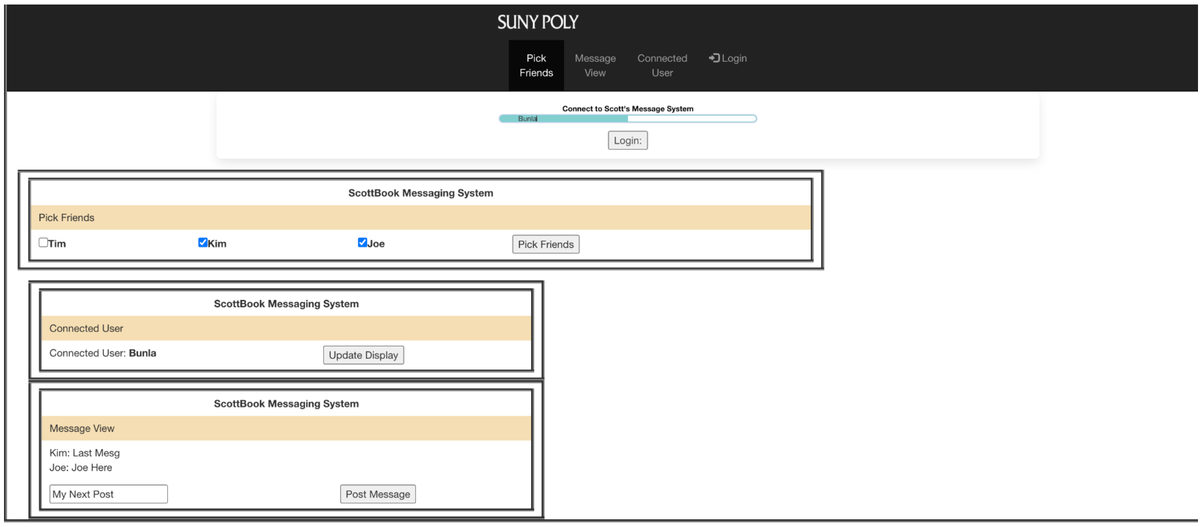
```
# input files for WebGL
# https://people.sc.fsu.edu/~jburkardt/data/tec/tec.html

  use strict;
  use warnings;
  use lib '/home/f/csci/scott/www/cgi-bin/PMS/Class-Accessor-0.34/blib/lib';
  use base 'Class::Accessor::Fast';
  use Class::Accessor::Fast;
  use Class::Accessor;
  use lib '/home/f/csci/scott/www/cgi-bin/PMS/CGI-Ajax-0.697/blib/lib';
  use CGI::Ajax;
  use lib '/home/f/csci/scott/www/cgi-bin/PMS/CGI-4.35/lib';
  use CGI;

  my %values;
   open (STDERR, ">&STDOUT");
   select(STDERR); $| = 1;
   select(STDOUT); $| = 1;
  #my $q = new CGI;
  my $q = CGI->new ;

  print "Content-type: text/html\n\n";
  #print $q->header('text/html'),
  #print "hello\n\n";
  #my @values  = $q->multi_param('QUERY_STRING');
   #$q->CGI::ReadParse(*values);

   my $myPickType = $q->CGI::param('mypicktype');

   my $myFunc = $q->CGI::param('myfunc');
   my $myRunid = $q->CGI::param('myrunid');



if ($myFunc eq "getpicks") {
if ( ( $myPickType eq "run" ) || ( $myPickType eq "display" )) {
  chdir("RUN351");
  my @picknames;
  $picknames[0] = $myPickType; # handler will use first name in the array as type for Picker List
  my $mycnt2 = 1;
  my $inputName2 = "";
  my $somedir2 = ".";
  opendir(DIR, $somedir2) || die "Can't open directory $somedir2: $!";
  my @tmpdat2 = readdir(DIR);
  foreach $inputName2 (@tmpdat2) {
#print "match $mycnt2 $inputName\n";
  #if ($inputName2 =~ m/\./ ) {
  if (!($inputName2 eq ".") && !($inputName2 eq "..") && !($inputName2 eq "files")){
        $picknames[$mycnt2++] = $inputName2;
  }
#print "found $inputName\n";
#print "\n";
  #
  }
  my $utf8_encoded_json_picknames = JSON::encode_json (\@picknames);
  print $utf8_encoded_json_picknames;
  exit;
}
}


if ( $myFunc eq "pickResults") {
}

# Process sendRequest1:
# Code from exampleWebGL.cgi
# myfunc=execexample&exampleTestCase=' + theTestCase

if ( $myFunc eq "execexample") {

  #require File::Temp;
  #use File::Temp ();
  #$dir = File::Temp->newdir('tempXXXXX');

  # Make a temporary directory for this run
  my $myTestCase = $q->CGI::param('exampleTestCase');
  chomp $myTestCase;

  my $range = 100000;
  my $dirname = int(rand($range));
  #$dirname =  "CS351_tmp" . $dirname . int(rand($range));
  $dirname =  $myTestCase . "/CS351_tmp" . $dirname ;
  mkdir $dirname, 0755;
  #print $dirname . "\n";

  chdir $dirname;
  # Link the data files to this directory
  `ln -s /home/f/csci/scott/www/cgi-bin/DATA/mydata.dat .`;
  `ln -s /home/f/csci/scott/www/cgi-bin/DATA/mydata.dat mydata2.dat`;
  `ln -s /home/f/csci/scott/www/cgi-bin/DATA/mydata.dat mydata3.dat`;

   #`/home/f/csci/scott/www/cgi-bin/TEC360/prog TEC360_inputs.inp >& myOutput`;

  my @dirnames;
  $dirnames[0] = $dirname; # RunID tells browser where the result is
  #$dirnames[1] = "SCOTT";
  my $mycnt1 = 1;
  my $inputName = "";
my $somedir = ".";
opendir(DIR, $somedir) || die "Can't open directory $somedir: $!";
my @tmpdats = readdir(DIR);

foreach $inputName (@tmpdats) {
#print "match $mycnt1 $inputName\n";
if ($inputName =~ m/.*dat$/ ) {
        $dirnames[$mycnt1++] = $inputName;
#print "found $inputName\n";
#print "\n";
}
}


  #my @myNames = exec("ls *.dat");
  #foreach $inputName (@myNames) {
  #foreach $inputName (`ls *.dat`) {
        #chomp $inputName;
        #$dirnames[0] += "++" + $inputName;
        #$dirnames[$mycnt1++] = $inputName;
  #}
  #print @dirnames;
  #exit;

  my $utf8_encoded_json_dirname = JSON::encode_json (\@dirnames);
  #my $utf8_encoded_json_dirname = JSON::encode_json (\@myNames);
  print $utf8_encoded_json_dirname;
  #my $json_text   = $JSON::json->encode( $dirname );
  #print $json_text ;
  exit;
}
```

```
# Encode Vertices and Faces to JSON
# imports encode_json, decode_json, to_json and from_json.
# imports encode_json, decode_json, to_json and from_json.
# simple and fast interfaces (expect/generate UTF-8)


 use constant { true => 1, false => 0 };

 use lib '/home/f/csci/scott/www/cgi-bin/JSON-2.90';
 use lib '/home/f/csci/scott/www/cgi-bin/JSON-2.90/lib';
 use lib::JSON;
 use JSON::backportPP;

my @jsonFaces;
my @jsonVertices;

sub doMyParse {


#print "HELLO1 \n";

open(tec360, "<mydata.dat") || die "Can't open tec360 dat: $!\n";

#print "HELLO2 \n";

my @newarray = <tec360>;
##print "NEWARRAY $newarray[0] \n";
##print "NEWARRAY $newarray[1] \n";
##print "NEWARRAY $newarray[2] \n";
##print "NEWARRAY $newarray[3] \n";
close (tec360);

my $foundtriangle = false;
my $foundvertex = false;

my $filename="myData";
my $fcnt = 0;
#foreach $line (@newarray) {
#
my $faceCnt = 0;
my $vertexCnt = 0;
my $line = 0;
my $filename2 = 0;
my @vals;

open(myOutfile, ">TEST28FEB2018") || die "Can't open $filename2 : $!\n";

my $cnt = 0;
while ($cnt <= $#newarray) {

        $line = $newarray[$cnt];
        if ( $line =~ /TITLE/ ) {
                $foundtriangle = false;
                #$line = <tec360>;
                #print "c360_surf $cnt  $newarray[$cnt++]; \n";
                $cnt++;
                #print "c360_surf $cnt  $newarray[$cnt++]; \n";
                $cnt++;
                #print "c360_surf $cnt  $newarray[$cnt++]; \n";
                $cnt++;
                $line = $newarray[$cnt++];
                #$line = <tec360>;
                #print "c360_surf $cnt  $line \n";
                #$line = <tec360>;
                #print "c360_surf $cnt  $line \n";
                if ($fcnt > 0) {
                        close($filename2);
                }
                $fcnt++;
                $filename2 = "$filename" . "$fcnt";

                #open(myOutfile, ">$filename2") || die "Can't open $filename2 : $!\n";
        }
        else {
                $cnt++;
        }

        chomp $line;
        #@vals = split(/ .+/, $line);
        $vals[4] = "ScottSpetka";
        @vals = split(/ {1,}/, $line);
        #if ($vals[4] == "") {}
        #if ($vals[4] eq "") {}
        if ( defined $vals[5] ) {
                if ($foundtriangle eq false) {
                        $foundtriangle = true;
                        print myOutfile "TRIANGLES\n";
                }
#geom.vertices.push( new THREE.Vector3( 10, -10, 0 ) );
#geom.faces.push( new THREE.Face3( 0, 1, 2 ) );

                print myOutfile " geom.vertices.push( new THREE.Vector3( $vals[1], $vals[2], $vals[3]
));\n";
                        $jsonVertices[$vertexCnt++] = $vals[1] . ";" . $vals[2] . ";" . $vals[3];
        }
        else {
                print myOutfile " geom.faces.push( new THREE.Face3( $vals[1], $vals[2], $vals[3]
));\n";
                        $jsonFaces[$faceCnt++] = $vals[1] . ";" . $vals[2] . ";" . $vals[3];
                        if (defined $vals[4] ) {
                print myOutfile " geom.faces.push( new THREE.Face3( $vals[2], $vals[3], $vals[4]
));\n";
                        $jsonFaces[$faceCnt++] = $vals[1] . ";" . $vals[3] . ";" . $vals[4];
                        }
        }
}

}  # end of sub doMyParse

if ( $myFunc eq "faces") {
 #chdir $myRunid;
 chdir "/home/f/csci/scott/www/cgi-bin/DATA";
 doMyParse();
 #print "FACES: ";
 #print @jsonFaces;
 #exit;
 my $utf8_encoded_json_faces = JSON::encode_json (\@jsonFaces);
 print $utf8_encoded_json_faces;
}

if ( $myFunc eq "vertices") {
 #chdir $myRunid;
 chdir "/home/f/csci/scott/www/cgi-bin/DATA";
 doMyParse();
 my $utf8_encoded_json_vertices = JSON::encode_json (\@jsonVertices);
 #$perl_hash_or_arrayref  = decode_json $utf8_encoded_json_text;
 #print "FACES: ", $utf8_encoded_json_faces;
 #print "<br>\n";
 #print "VERTICES ", $utf8_encoded_json_vertices;
 #print "<br>\n";
 print $utf8_encoded_json_vertices;
```

```
}

 print "Content-type: text/html\n\n";
  my $q = CGI->new;
  my $value = $q->param('myfunc');
  my $dir = 'cgi-bin/post';
  if($value eq 'updateDisplay'){
      my $name = $q->param('name');
      open(POST,$dir);
      my @lines = <POST>;
      my $out = '';
      foreach my $line (@lines){
          my @everything = split(",",$line);
          if($everything[0] eq $name){
              $out .= $line .= ";" ;
          }
      }
      close(POST);
      print $out;
  }

  if($value eq "topicDisplay"){
      my $topics = $q->param('topics');
      my @topic = split(",",$topics);
      open(POST,$dir);
      my @lines = <POST>;
      my $out = '';
      foreach my $line (@lines){
          my @everything = split(",",$line);
          foreach my $topicd (@topic){
              if($topicd eq $everything[1]){
              $out .= $line .= ";" ;
              }
          }
      }
      close(POST);
      print $out;
  }

  if($value eq "postIdea"){
      my $name = $q->param('name');
      my $topic = $q->param('topics');
      my $stuff = $q->param('stuff');
      open(POST,">>",$dir);
```

```
            print $topic;
        my $newIdea = "\n" . $name. "," . $topic . "," . $stuff;
        print POST $newIdea;
        my $out= 1;
        print $out;
    }




#Routing
{
    package MyWebServer;

    use HTTP::Server::Simple::CGI;

    use base qw(HTTP::Server::Simple::CGI);

    my %dispatch = (
        '/index.html' => \&resp_index,
        # ...
    );

    sub handle_request {
        my $self = shift;
        my $cgi  = shift;

        my $path = $cgi->path_info();
        my $handler = $dispatch{$path};
        if (ref($handler) eq "CODE") {
            print "HTTP/1.0 200 OK\r\n";
            $handler->($cgi);
        } else {
            print "HTTP/1.0 404 Not found\r\n";
            print $cgi->header,
                $cgi->start_html('Not found'),
                $cgi->h1('Not found'),
                $cgi->end_html;
        }
    }

    sub resp_index {
        my $cgi  = shift;   # CGI.pm object
        return if !ref $cgi;
        my $who = $cgi->param('name');

        print $cgi->header,
            $cgi->start_html("index"),
            $cgi-h1("THIS IS INDEX"),
            $cgi->end_html;
    }
}

my $pid = MyWebServer->new()->background();
print "Use 'kill $pid' to stop server.\n";
```

Screenshot



ScottBook Messaging System

Pick Friends

☐ Tim ☑ Kim ☑ Joe [Pick Friends]

ScottBook Messaging System

Connected User

Connected User: **Bunla** [Update Display]

ScottBook Messaging System

Message View

Kim: Last Mesg
Joe: Joe Here

My Next Post [Post Message]