

```
1 <!-->
2 index.html
3 <-->
4
5 <html ng-app="profilePage">
6
7     <head>
8         <title>Function</title>
9         <link rel="icon" type="image/png" href="favicon-32x32.png"
•             sizes="32x32" />
10        <link href="movement.css" rel="stylesheet" type="text/css"></
•             link>
11        <script type="text/javascript" src="js/angular-min.js"></
•             script>
12        <script type="text/javascript" src="js/angular-
•             cookies.min.js"></script>
13        <script type="text/javascript" src="js/md5-min.js"></script>
14        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
15            integrity="sha384-q8i/
•                 X+965Dz00rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6ji
•                 zo"
16            crossorigin="anonymous"></script>
17        <link rel="stylesheet" href="https://fonts.googleapis.com/
•                 icon?family=Material+Icons"></link>
18        <script type="text/javascript" src="js/profile.js"></script>
19     </head>
20
21     <body ng-controller="profilePageController">
22         <div id="login" class="loginContainer">
23             <input ng-model="emailOrUsername" class="loginInput"
•                 type="text" placeholder="Email/Username">
24             <input ng-model="loginPassword" class="loginInput"
•                 type="password" placeholder="password">
25             <div class="loginButton" ng-click="loginSubmit()">Login</
•                 div>
26             <div class="rahText" ng-click="swapTo('signup')">New
•                 around here? Create an account!</div>
27
28         </div>
29
30         <div id="signup" class="loginContainer">
31             <input ng-model="username" class="loginInput" type="text"
•                 placeholder="username">
32             <input ng-model="email" class="loginInput" type="text"
•                 placeholder="email">
```

```
password = email /  
33    <input ng-model="password" class="loginInput"  
•        type="password" placeholder="password">  
34    <div class="loginButton" ng-click="signupSubmit()">Sign  
•        Up</div>  
35    <div class="rahText" ng-click="swapTo('login')">Already  
•        have an account? Log in!</div>  
36    </div>  
37  
38    </body>  
39 </html>  
40
```

```
1  /*
2   profile.js
3  */
4
5  var app = angular.module("profilePage", ["ngCookies", "angular-
• md5"]); //
6
7  app.controller("profilePageController", ["$scope", "$http",
• "$window", "$cookies", "md5", function($scope, $http, $window,
• $cookies, md5) {
8      $("#signup").hide();
9
10     /////////////////////////////////
•     /////////////////////
11     //
12     //  LOGIN
13     //
14
15     $scope.loginSubmit = function () {
16         if(!$scope.emailOrUsername || !$scope.loginPassword) {
17             alert("Please enter a Password");
18             return;
19         }
20
21         var params = {
22             action: "checkProfile",
23             emailOrUsername: $scope.emailOrUsername,
24             password: md5.createHash($scope.loginPassword)
25         };
26
27         $http.get("profileManager", { params: params
• }).then(function(res) {
28             // success
29             if (res.data.passwordOk) {
30                 console.log("Correct Password");
31                 window.location.href = "main.html";
32             }
33             else {
34                 alert("Incorrect Username/Password");
35             }
36         }, function() {
37             // failure
38         });
39     };
40
```

```

40
41     ///////////////////////////////////////////////////////////////////
42     //
43     //  SIGNUP
44     //
45
46     $scope.signupSubmit = function () {
47         if(!$scope.username || !$scope.email || !$scope.password) {
48             alert("bad stuff");
49             return;
50         }
51
52         var params = {
53             action: "addProfile",
54             name: $scope.name,
55             username: $scope.username,
56             email: $scope.email,
57             password: md5.createHash($scope.password)
58         };
59         //console.log(params);
60         $http.get("profileManager", { params: params
61         }).then(function(res) {
62             //success
63             if (res.data.newUser) {
64                 window.location.href = "main.html";
65             }
66             else {
67                 alert("Username/Email already in use!");
68             }
69         }, function() {
70             // failure
71         });
72     };
73
74     $scope.swapTo = function(newForm) {
75         $(".loginContainer").hide();
76         $("#" + newForm).show();
77     };
78     ///////////////////////////////////////////////////////////////////
79     //
80     //  CHANGE PASSWORD
81     //

```

```
82
83     $scope.changePassword = function() {
84         if(md5.createHash($scope.oldPassword) ===
85             • $cookies.get("password")) {
86             console.log("yo im here");
87             var params = {
88                 action: "changePassword",
89                 username: $cookies.get("username"),
90                 password: md5.createHash($scope.newPassword)
91             };
92             //console.log(params);
93             $http.get("profileManager", { params: params
94             • }).then(function(res) {
95                 //success
96                 if(res.data.passwordOk) {
97                     window.location.href = "myPage.html";
98                     console.log("going back");
99                 }
100             }, function() {
101                 // failure
102             });
103         };
104         $scope.backToMainPage = function() {
105             window.location.href = "myPage.html";
106         };
107
108
109
110     });
111
```

```
1 <!-->
2 main.html
3 <-->
4
5 <html ng-app="myPage">
6
7     <head>
8         <title>Function</title>
9         <link rel="icon" type="image/png" href="favicon-32x32.png"
•             sizes="32x32" />
10        <link href="movement.css" rel="stylesheet" type="text/css"></
•            link>
11        <script type="text/javascript" src="js/jquery-min.js"></
•            script>
12        <script type="text/javascript" src="js/angular-min.js"></
•            script>
13        <script type="text/javascript" src="js/angular-
•            cookies.min.js"></script>
14        <script type="text/javascript" src="js/md5-min.js"></script>
15        <link rel="stylesheet" href="https://fonts.googleapis.com/
•            icon?family=Material+Icons"></link>
16        <script type="text/javascript" src="js/myPageManager.js"></
•            script>
17    </head>
18
19    <body ng-controller="mainController">
20        <div class="dim" ng-click="hideAll()"></div>
21        <div class="projectDropdown">
22            {{currentProjectName}}
23            <input class="loginInput" ng-model="newProjectName"
•                style="background-color: #E9E9E9!important"
•                placeholder="Project Name" />
24            <div class="submitButton" ng-
•                click="submitProjectChanges()">
25                Save Changes
26            </div>
27        </div>
28        <div class="stageDropdown">
29            {{currentStageName}}
30            <input class="loginInput" ng-model="newStageName"
•                style="background-color: #E9E9E9!important"
•                placeholder="New Stage Name" />
31            <div class="submitButton" ng-click="submitStageChanges()">
32                Save Changes
33        </div>
```

```
33      >/< />
34  </div>
35  <div class="storyDropdown">
36      {{currentStoryName}}
37      <input class="loginInput" ng-model="newStoryName"
38          style="background-color: #E9E9E9!important"
39          placeholder="New Story Name" />
40      <div class="submitButton" ng-click="submitStoryChanges()">
41          Save Changes
42      </div>
43  </div>
44
45  <div id="header">
46      DeleGator
47  </div>
48
49  <div class="cardContainer">
50      <div id="project{{project.serial}}" class="card" ng-
51          repeat="project in projects"
52          ng-click="getStages(project.serial)">
53          {{project.name}}
54          <i class="material-icons editIcon" ng-
55              click="openProjectDropdown(project)">edit</i>
56      </div>
57      <div class="submitButton" ng-
58          click="openProjectDropdown()">
59          Create Project
60      </div>
61  </div>
62
63  <div class="stagesContainer">
64      <div class="stage" ng-repeat="stage in stages" ng-
65          init="getStories(stage, $index)" ng-
66          mouseup="dropStory(stage)">
67          {{stage.name}}
68          <div class="story" ng-repeat="story in
69              stories[$index]" ng-mousedown="moveStory(story)">
70              {{story.name}}
71              <i class="material-icons editIcon" ng-
72                  click="openStoryDropdown(stage.serial,
73                  story)">edit</i>
74          </div>
75          <div class="addStory" ng-
76              click="openStoryDropdown(stage.serial)">
77              Add a story
78      </div>
79  </div>
```

```
67          </div>
68      </div>
69  </div>
70
71  <div class="fob" ng-click="openStageDropdown()">
72      <i class="material-icons">add</i>
73  </div>
74
75  <div id="tempStory">
76      {{currentStory.name}}
77  </div>
78
79  </body>
80 </html>
81
```

```

1  /*
2   myPageManager.js
3  */
4
5  var app = angular.module("myPage", ["ngCookies"]);
6
7  app.controller("mainController", ["$scope", "$http", "$window",
8  • "$cookies", "$compile", function($scope, $http, $window, $cookies,
9  • $compile) {
10
11     $(".dim").hide();
12     $(".projectDropdown").hide();
13     $(".stageDropdown").hide();
14     $(".storyDropdown").hide();
15
16     /////////////////
17     //
18     //  FETCHING FUNCTIONS
19     //
20
21     $scope.stories = [];
22     var currentProject = -1;
23     function refreshProjects() {
24         var params = {
25             action: "getProjects",
26             username: $cookies.get("username"),
27         };
28         $http.get("myPageManager", { params: params
29         }).then(function(res) {
30             //success
31             $scope.projects = res.data.rows;
32             manuallyGetStages(res.data.rows[0].serial);
33             $("#project" +
34                 res.data.rows[0].serial).addClass("selectedProject");
35         }, function() {
36             // failure
37             console.log("FAIL");
38         });
39     }
40     refreshProjects();
41
42
43     $scope.getStages = function(serial) {
44         currentProject = serial;
45
46         manuallyGetStages(serial);
47
48         $http.get("myPageManager", { params: params
49         }).then(function(res) {
50             $scope.stories = res.data;
51         }, function() {
52             console.log("FAIL");
53         });
54     }
55
56     $scope.$watch("currentProject", function(newVal, oldVal) {
57         if (newVal > -1) {
58             manuallyGetStages($scope.projects[newVal].serial);
59         }
60     });
61
62     $scope.$watch("stories", function(newVal, oldVal) {
63         if (newVal.length > 0) {
64             $scope.currentStory = newVal[0];
65         }
66     });
67
68     $scope.$watch("currentStory", function(newVal, oldVal) {
69         if (newVal) {
70             manuallyGetStory(newVal.serial);
71         }
72     });
73
74     $scope.$watch("currentStage", function(newVal, oldVal) {
75         if (newVal) {
76             manuallyGetStage(newVal.serial);
77         }
78     });
79
80     $scope.$watch("currentStage", function(newVal, oldVal) {
81         if (newVal) {
82             manuallyGetStory(newVal.serial);
83         }
84     });
85
86     $scope.$watch("currentStory", function(newVal, oldVal) {
87         if (newVal) {
88             manuallyGetStage(newVal.currentStage);
89         }
90     });

```

```

40     currentProject = serial,
41     $(".card").removeClass("selectedProject");
42     $("#project" + currentProject).addClass("selectedProject");
43     var params = {
44         action: "getStages",
45         projectSerial: serial
46     };
47     $http.get("myPageManager", { params: params
48     }).then(function(res) {
49         //success
50         $scope.stages = res.data.rows;
51     }, function() {
52         // failure
53         console.log("FAIL");
54     });
55
56     $scope.getStories = function(stage, stageIndex) {
57         //console.log(stage);
58     var params = {
59         action: "getStories",
60         stageSerial: stage.serial
61     };
62     $http.get("myPageManager", { params: params
63     }).then(function(res) {
64         //success
65         $scope.stories[stageIndex] = res.data.rows;
66     }, function() {
67         // failure
68         console.log("FAIL");
69     });
70
71     /////////////////////////////////
72     ///////////////////////////////
73     // DRAG AND DROP FUNCTIONALITY
74     //
75
76     var oldX;
77     $("body").mousemove(function(e) {
78         //console.log(e);
79         var degree = (e.pageX - oldX);
80         if($scope.currentStory !== 0) {
81             $("#tempStory").css({

```

```

82             "top": e.pageY,
83             "left": e.pageX,
84             WebkitTransform: 'rotate(' + degree + 'deg)',
85             opacity: 0.6
86         });
87     }
88     oldX = e.pageX;
89 });
90
91 $scope.currentStory = 0;
92 $scope.moveStory = function(story) {
93     $scope.currentStory = story;
94     $("body").css({"user-select": "none"});
95 };
96 $scope.dropStory = function(stage) {
97     if($scope.currentStory !== 0) {
98         $("#tempStory").css({opacity: 0});
99         $("body").css({"user-select": "auto"});
100
101     var params = {
102         action: "changeStory",
103         storySerial: $scope.currentStory.serial,
104         newStageSerial: stage.serial
105     };
106     $http.get("myPageManager", { params: params
107     }).then(function(res) {
108         //success
109         if(res.data.success) {
110             manuallyGetStages(currentProject);
111         }
112     }, function() {
113         // failure
114         console.log("FAIL");
115     });
116     $scope.currentStory = 0;
117 }
118
119 function manuallyGetStages(projectSerial) {
120     $(".card").removeClass("selectedProject");
121     $("#project" + projectSerial).addClass("selectedProject");
122     var params = {
123         action: "getStages",
124         projectSerial: projectSerial
125     };

```

```
125        ,
126        $http.get("myPageManager", { params: params
127      •       }).then(function(res) {
128          //success
129          $scope.stages = res.data.rows;
130      }, function() {
131          // failure
132          console.log("FAIL");
133      });
134
135      /////////////////////////////////
136      /////////////////////
137      // PROJECT-RELATED FUNCTIONS
138      //
139
140      $scope.openProjectDropdown = function(project) {
141          if(project) {
142              $scope.currentProjectName = "Edit project '" +
143              •           project.name + "'";
144          } else {
145              $scope.currentProjectName = "Create a Project";
146          }
147          $(".dim").fadeIn(200);
148          $(".projectDropdown").fadeIn(200);
149      };
150
151      $scope.hideAll = function() {
152          $(".dim").fadeOut(200);
153          $(".projectDropdown").fadeOut(200);
154          $(".stageDropdown").fadeOut(200);
155          $(".storyDropdown").fadeOut(200);
156      };
157
158      $scope.submitProjectChanges = function() { //Fuction for
159          •           creating and editing projects
160          var params;
161          if($scope.currentProjectName === "Create a Project") {
162              params = {
163                  action: "createProject",
164                  username: $cookies.get("username"),
165                  projectName: $scope.newProjectName
166              };
167          } else {
```

```

166         params = {
167             action: "editProject",
168             projectSerial: currentProject,
169             newProjectName: $scope.newProjectName
170         };
171     }
172     $http.get("myPageManager", { params: params
173     • }).then(function(res) {
174         //success
175         if(res.data.success) {
176             $(".dim").fadeOut(200);
177             $(".projectDropdown").fadeOut(200);
178             refreshProjects();
179         },
180         function() {
181             // failure
182             console.log("FAIL");
183         });
184     };
185     /////////////////////////////////
186     /////////////////////
187     // STAGE-RELATED FUNCTIONS
188     //
189
190     $scope.openStageDropdown = function(stage) {
191         $scope.currentStageName = "Add a Stage";
192         $(".dim").fadeIn(200);
193         $(".stageDropdown").fadeIn(200);
194     };
195
196     $scope.submitStageChanges = function() { //Fuction for creating
197     • and editing stages
198         var params = {
199             action: "createStage",
200             projectSerial: currentProject,
201             newStageName: $scope.newStageName
202         };
203         $http.get("myPageManager", { params: params
204         • }).then(function(res) {
205             //success
206             if(res.data.success) {
207                 $(".dim").fadeOut(200);
208                 $(".stageDropdown").fadeOut(200);

```

```

207             manuallyGetStages(currentProject);
208         }
209     }, function() {
210         // failure
211         console.log("FAIL");
212     });
213 };
214
215 ///////////////////////////////////////////////////////////////////
216 //
217 // STORY-RELATED FUNCTIONS
218 //
219
220 var currentStageSerial;
221 var currentStorySerial;
222 $scope.openStoryDropdown = function(stageSerial, story) {
223     currentStageSerial = stageSerial;
224     if(story) {
225         $scope.currentStoryName = "Edit story '" + story.name +
226         "'";
227         currentStorySerial = story.serial;
228     } else {
229         $scope.currentStoryName = "Add a Story";
230     }
231     $(".dim").fadeIn(200);
232     $(".storyDropdown").fadeIn(200);
233 };
234
235 $scope.submitStoryChanges = function() { //Fuction for creating
236     • and editing stories
237     console.log("REEEE");
238     var params;
239     if($scope.currentStoryName === "Add a Story") {
240         params = {
241             action: "createStory",
242             stageSerial: currentStageSerial,
243             newStoryName: $scope.newStoryName
244         };
245     } else {
246         params = {
247             action: "editStory",
248             storySerial: currentStorySerial,
249             newStoryName: $scope.newStoryName
250         };

```

```
248         };
249     }
250     $http.get("myPageManager", { params: params
251   • }).then(function(res) {
252       //success
253       if(res.data.success) {
254           $(".dim").fadeOut(200);
255           $(".storyDropdown").fadeOut(200);
256           manuallyGetStages(currentProject);
257       }
258   }, function() {
259       // failure
260       console.log("FAIL");
261   });
262
263 });
264
```

```
1  /*
2   movement.css
3  */
4
5  @font-face {
6      font-family: "myFont";
7      src: url("fonts/Montserrat-SemiBold.ttf");
8  }
9 /*
10 ::-webkit-scrollbar {
11     width: 0px; /* remove scrollbar space */
12     background: transparent; /* optional: just make scrollbar
13     invisible */
14 }
15 /* optional: show position indicator in red */
16 ::-webkit-scrollbar-thumb {
17     background: #FF0000;
18 }
19 body {
20     margin: 0;
21     padding: 0;
22     /*color: #dcfce3!important;
23     background-color: #1a1a20;*/
24     background-color: #e0e0e0;
25     color: #666!important;
26     display: block;
27     text-align: center;
28     overflow-y: hidden;
29     font-family: "myFont"!important;
30     font-size: 10pt!important;
31     transition: all 0.8s;
32 }
33
34 input:focus {
35     outline: none;
36 }
37
38 .editIcon {
39     font-size: 14pt!important;
40     float: right;
```

```
41     opacity: 0;
42     transition: 0.2s;
43 }
44
45 .dim {
46     position: absolute;
47     z-index: 5000;
48     background-color: black;
49     opacity: 0.4;
50     width: 100%;
51     height: 100%;
52 }
53
54 .projectDropdown {
55     position: absolute;
56     left: 0;
57     right: 0;
58     margin: auto;
59     margin-top: 100px;
60     z-index: 5001;
61     width: 384px;
62     box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
63     border-radius: 0.8rem;
64     background-color: #FFF;
65     padding: 20px;
66 }
67 .stageDropdown {
68     position: absolute;
69     left: 0;
70     right: 0;
71     margin: auto;
72     margin-top: 100px;
73     z-index: 5001;
74     width: 384px;
75     box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
76     border-radius: 0.8rem;
77     background-color: #FFF;
78     padding: 20px;
79 }
80 .storyDropdown {
81     position: absolute:
```

```
--      -----
82      left: 0;
83      right: 0;
84      margin: auto;
85      margin-top: 100px;
86      z-index: 5001;
87      width: 384px;
88      box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
89      border-radius: 0.8rem;
90      background-color: #FFF;
91      padding: 20px;
92  }
93
94
95 .loginInput {
96     display: block;
97     background-color: #ffffff!important;
98     margin: 0 auto!important;
99     margin-top: 8px!important;
100    width: 320px!important;
101    padding: 18px!important;
102    transition: all 0.3s;
103    z-index: 1!important;
104    position: relative;
105    border: 0!important;
106    border-radius: 0.8em;
107    font-size: 12pt;
108  }
109 .loginInput:focus {
110     background-color: #ffffff!important;
111     width: 380px!important;
112     border-bottom: 0!important;
113     -webkit-box-shadow: 0 2px 48px rgba(0,0,0,0.1)!important;
114     box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
115     z-index: 2!important;
116  }
117 .loginButton {
118     font-size: 14px;
119     background: #2467fc;
120     border: none;
121     border-radius: .8em;
122     color: #FFF.
```

```
122     color: #fff,
123     width: 294px;
124     text-transform: uppercase;
125     font-weight: 700;
126     letter-spacing: .1em;
127     padding: 15px 17px;
128     line-height: 16px;
129     position: relative;
130     margin: auto;
131     margin-top: 24px;
132     margin-bottom: 0;
133     transition: all .2s;
134 }
135 .loginButton:hover {
136     margin-top: 20px;
137     margin-bottom: 4px;
138     cursor: pointer;
139     box-shadow: 0 2px 24px rgba(0,0,0,0.2);
140 }
141 .rahText {
142     margin-top: 26px;
143     font-size: 12pt;
144     cursor: pointer;
145 }
146
147 .submitButton {
148     font-size: 14px;
149     background: #2467fc;
150     border: none;
151     border-radius: .8em;
152     color: #fff;
153     width: 70%;
154     text-transform: uppercase;
155     font-weight: 700;
156     letter-spacing: .1em;
157     padding: 15px 17px;
158     line-height: 16px;
159     position: relative;
160     margin: auto;
161     margin-top: 24px;
162     transition: all .2s;
163 }
```

```
163 }
164 .submitButton:hover {
165     margin-top: 20px;
166     cursor: pointer;
167 }
168 .fob {
169     background: #2467fc;
170     border: none;
171     font-size: 16pt;
172     border-radius: 50px;
173     color: #fff;
174     width: auto;
175     height: auto;
176     padding: 18px;
177     position: fixed;
178     z-index: 5;
179     bottom: 80px;
180     right: 80px;
181     transition: all .2s;
182 }
183 .fob:hover {
184     cursor: pointer;
185     bottom: 84px;
186     box-shadow: 0 2px 6px rgba(0,0,0,0.7)!important;
187 }
188
189 #login, #signup {
190     left: 0;
191     right: 0;
192     text-align: center;
193     margin: auto;
194     width: auto;
195     padding: 60px;
196 }
197
198
199 #header {
200     top: 0;
201     color: #eee!important;
202     width: 100%;
203     margin-left: auto;
```

```
204     margin-right: auto;
205     position: fixed;
206     background-color: #141418;
207     z-index: 8;
208     padding: 16px;
209     text-align: center;
210     height: auto;
211     font-weight: 0.1;
212     box-shadow: 0 1px 1px rgba(0, 0, 0, 0.6)
213 }
214
215 .cardContainer {
216     left: 0;
217     top: 0;
218     float: left;
219     bottom: 0;
220     margin: auto;
221     width: 18%;
222     height: auto;
223     padding-top: 86px;
224     position: fixed;
225
226     background-color: #eeeeee;
227     border: solid 0px #ccc;
228
229     -webkit-box-shadow: 0 2px 48px rgba(0,0,0,0.1)!important;
230     box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
231 }
232
233 .card {
234     font-family: "myFont";
235     display: block;
236     margin: 0 0!important;
237     margin-top: 0px!important;
238     width: 70%!important;
239     padding: 20px 16px;
240     transition: all 0.3s;
241     z-index: 1!important;
242     left: 0!important;
243     border: 0!important;
244     border-top-right-radius: .8em;
```

```
245     border-bottom-right-radius: .8em;
246     text-align: left;
247 }
248 .card:hover {
249     cursor: pointer;
250     background-color: #ffffff!important;
251     width: 80%!important;
252     border-bottom: 0!important;
253     -webkit-box-shadow: 0 2px 48px rgba(0,0,0,0.1)!important;
254     box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
255     z-index: 5!important;
256 }
257
258 .card:hover i {
259     opacity: 1;
260 }
261
262
263
264 .stagesContainer {
265     height: 90%;
266     width: auto;
267     position: absolute;
268     right: 0;
269     left: 18%;
270     padding-left: 2%;
271     margin: auto;
272     padding-top: 60px;
273     overflow-x: auto;
274 }
275 .stage {
276     /*background-color: #141418;*/
277     border-radius: 1em;
278     padding: 1.125em;
279     height: 30em!important;
280     width: 280px;
281     float: left;
282     display: inline-block!important;
283     margin: 24px;
284
285     background-color: #e5e5e5;
```

```
286     border: solid 1px #ccc;
287 }
288 .story {
289     display: block;
290     width: auto;
291     height: auto;
292     padding: 1.2em;
293     font-size: 11pt!important;
294     line-height: 1;
295     margin-top: 24px;
296     color: #666666;
297     /*background-color: #222222b;*/
298     background-color: rgb(212,214,217);
299     background-clip: padding-box;
300     border: 0 solid #ced4da;
301     border-radius: .25em;
302     transition: .2s linear;
303     text-align: left;
304     cursor: pointer;
305 }
306 .story:hover {
307     box-shadow: 0 2px 6px rgba(0,0,0,0.3)!important;
308 }
309
310 .story:hover .editIcon {
311     opacity: 1;
312 }
313
314 #tempStory {
315     position: absolute;
316     z-index: 1000;
317     width: calc(280px - 2.25em);
318     height: auto;
319     padding: 1.2em;
320     font-size: 11pt!important;
321     line-height: 1;
322     color: #666666;
323     background-color: rgb(212,214,217);
324     background-clip: padding-box;
325     border: 0 solid #ced4da;
326     border-radius: .25em;
```

```
--> .addStory {
327    text-align: left;
328    cursor: pointer;
329    pointer-events: none;
330    box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
331    opacity: 0;
332 }
333
334 .addStory {
335     display: block;
336     width: auto;
337     height: auto;
338     padding: 1.2em;
339     font-size: 10pt!important;
340     line-height: 1;
341     margin-top: 24px;
342     color: #666666;
343     /*background-color: #22222b;*/
344     background-color: rgb(232,234,237);
345     background-clip: padding-box;
346     border: 2px dashed #ced4da;
347     border-radius: .25em;
348     transition: .15s linear;
349     text-align: left;
350     cursor: pointer;
351 }
352 .addStory:hover {
353     background-color: rgb(222,224,227)!important;
354 }
355
356
357 .selectedProject {
358     cursor: pointer;
359     background-color: #ffffff!important;
360     width: 80%!important;
361     -webkit-box-shadow: 0 2px 48px rgba(0,0,0,0.1)!important;
362     box-shadow: 0 2px 24px rgba(0,0,0,0.2)!important;
363     z-index: 5!important;
364     border-left: solid 4px #2467fc!important;
365 }
366
367 <input s
```

```
297 .input {
368     padding: 12px;
369     border-radius: 30px;
370     width: auto;
371     border: none;
372     margin: 20px;
373     margin-left: auto;
374     margin-right: auto;
375     left: auto;
376     right: auto;
377     display: flex;
378     outline: none;
379     border-style: solid;
380     border-width: 1px;
381     border-color: #3c4146;
382     display: flex;
383     transition: all 0.3s cubic-bezier(.25,.8,.25,1);
384 }
385 .input:focus {
386     box-shadow: 0 2px 4px grey;
387 }
388
389 .button {
390     background-color: #3c4146;
391     color: #EEE;
392     width: auto;
393     padding: 12px;
394     border-radius: 30px;
395     margin: 20px;
396     margin-left: auto;
397     margin-right: auto;
398     text-align: center!important;
399     display: flex;
400     left: auto;
401     right: auto;
402     cursor: pointer;
403     box-shadow: 0 8px 0 black;
404     transition: all 0.3s cubic-bezier(.25,.8,.25,1);
405     user-select: none;
406 }
407 .button:hover {
```

```
408     box-shadow: 0 4px 0 black;
409     margin-top: 24px;
410     margin-bottom: 16px;
411 }
412
413 .button:active {
414     transition: all 0;
415     background-color: white!important;
416     color: black!important;
417 }
418
419 .actionButton {
420     display: inline-flex;
421     z-index: 2;
422     opacity: 0;
423     margin-right: 40px;
424     margin-top: -90px!important;
425     transition: all 0.6s;
426     margin-right: -50px!important;
427 }
428
```

```
1  /*
2   server.js
3  */
4
5 // The one and only server function
6 /*jshint esversion: 6 */
7
8 exports.server = function() {
9
10    var express = require("express"),
11        app = express(),
12        mysql = require("mysql"),
13        cookie = require('cookie'),
14        md5 = require("md5"),
15        port = 81;
16
17    var pool = mysql.createPool({
18        connectionLimit: 5,
19        host: "localhost",
20        user: "kane",
21        password: "kanos999",
22        database: "mindmappr",
23        timezone: "utc"
24    });
25
26    var nodesArray = [];
27
28    function dbQuery(sql, callback) {
29        pool.getConnection((err, connection) => {
30            if (!err) {
31                connection.query(sql, function(err, rows) {
32                    if (!err) {
33                        callback(null, rows);
34                    }
35                    else {
36                        callback(err);
37                    }
38                });
39                connection.release();
40            }
41        });
42    }
43
44    /////////////////////////////////
45    /////////////////////////////////
```

```
  ///////////////
45  //
46  //  LOGIN + SIGNUP FUNCTIONS
47  //
48
49  app.get("/profileManager", (req, res) => {
50      var action = req.query.action,
51          sql;
52
53      switch (action) {
54          case "checkProfile":
55              sql = `SELECT * from users WHERE (username =
56  •      '${req.query.emailOrUsername}' OR email =
57  •      '${req.query.emailOrUsername}') AND password =
58  •      '${req.query.password}'`;
59              dbQuery(sql, function(err, rows) {
60                  var result = { passwordOk: false };
61                  if (rows.length) {
62                      result.passwordOk = true;
63                      res.cookie("username", rows[0].username);
64                      res.cookie("password", rows[0].password);
65                  }
66                  res.json(result);
67              });
68          break;
69
70          case "addProfile":
71              sql = `SELECT * from users WHERE username =
72  •      '${req.query.username}' OR email = '${req.query.email}'`;
73              dbQuery(sql, function(err, rows) {
74                  var result = { newUser: true };
75                  if (rows.length) {
76                      result.newUser = false;
77                  }
78                  else {
79                      sql = `INSERT INTO users (username, email,
80                            password) VALUES ('${req.query.username}',
81                            '${req.query.email}', '${req.query.password}')`;
82                      dbQuery(sql, function(err, rows) {});
83                      res.cookie("username", req.query.username);
84                      res.cookie("password", req.query.password);
85                  }
86                  res.json(result);
87              });
88          break;
89      }
90  });
91
92
```

```

83
84     case "changePassword":
85         res.cookie("password", req.query.password);
86         sql = `UPDATE users SET password =
87             •      '${req.query.password}' WHERE username =
88             •      '${req.query.username}'`;
89         dbQuery(sql, function(err, rows) {});
90         res.json({passwordOk:true});
91         break;
92     }
93     ///////////////////////////////////////////////////////////////////
94     ///////////////////////////////////////////////////////////////////
95     // MAIN PAGE FUNCTIONS
96     //
97
98     app.get("/myPageManager", (req, res) => {
99         var action = req.query.action,
100            sql;
101
102         switch (action) {
103             case "getProjects":
104                 sql = `SELECT * from projects WHERE owner =
105                     •      '${req.query.username}'`;
106                 dbQuery(sql, function(err, rows) {
107                     res.json({
108                         rows: rows
109                     });
110                 break;
111
112             case "createProject":
113                 sql = `INSERT INTO projects (name, owner) VALUES
114                     •      ('${req.query.projectName}', '${req.query.username}')`;
115                 dbQuery(sql, function(err, rows) {
116                     sql = `SELECT * from projects WHERE owner =
117                         •      '${req.query.username}'`;
118                     dbQuery(sql, function(err, rows) {
119                         var newProjectSerial = rows[rows.length-
120                             1].serial;
121                         sql = `INSERT INTO stages (name, projectSerial)
122                             VALUES ('To Do', '${newProjectSerial}'),
123                             ('Doing', '${newProjectSerial}'), ('Review',
124

```



```
156         break;
157
158     case "changeStory":
159         sql = `UPDATE stories SET stageSerial =
160 •           '${req.query.newStageSerial}' WHERE serial =
161 •           '${req.query.storySerial}'`;
160         dbQuery(sql, function(err, rows) {
161             res.json({success: true});
162         });
163         break;
164
165     case "editStory":
166         sql = `UPDATE stories SET name =
167 •           '${req.query.newStoryName}' WHERE serial =
168 •           '${req.query.storySerial}'`;
167         dbQuery(sql, function(err, rows) {
168             //console.log(sql);
169             res.json({success: true});
170         });
171         break;
172
173     case "editProject":
174         sql = `UPDATE projects SET name =
175 •           '${req.query.newProjectName}' WHERE serial =
176 •           '${req.query.projectSerial}'`;
175         dbQuery(sql, function(err, rows) {
176             res.json({success: true});
177         });
178         break;
179     }
180 });
181
182 // Where static content is served from.
183 app.use(express.static(__dirname + "/../static/", { index:
184 •   "index.html" }));
185
186 // Start server
187 app.listen(port);
188 console.log("Starting server on port", port);
189 };
```