

```

1 // Module: IT8904 BDD
2 // Assignment 1 - CA1
3 // Student No: P7461072
4 // Student Name: ONG WEI CHUAN
5
6 // Filename: app.js
7 // Directory: P7461072_BDD_CA1/controller/
8 // This file is under the controller layer
9 // Browser(View) <--> Controller <--> Model
10 // This file is to provide the web service to communicate
11 // with the browser (client)
12
13 // References
14 // 1. en.wikipedia.org/wiki/Film_genre
15 // 2. https://www.rottentomatoes.com/
16
17 https: var express = require("express");
18 var bodyParser = require("body-parser");
19 var user = require("../model/user");
20 var movie = require("../model/movie");
21 var genre = require("../model/genre");
22
23 var app = express(); //start express server
24
25 // POST/PUT/DELETE methods require body-parser
26 var urlencodedParser = bodyParser.urlencoded({ extended: false });
27 app.use(urlencodedParser); //attach body-parser middleware
28 app.use(bodyParser.json()); //parse json data
29
30 // To verify admin's credentials using email and password
31 // Once valid email and password are found in the database,
32 // the role of this login person must be an admin
33 // POST /login - To validate login by admin
34 app.post("/login", function (req, res) {
35     var email = req.body.email; //get email
36     var password = req.body.password; //get password
37
38     console.log("body.email = " + email);
39     console.log("body.password = " + password);
40
41     user.loginUser(email, password, function (err, results) {
42         // Error handling 1: If server error
43         if (err) {
44             res.status(500);
45             res.type("json");
46             res.send(`{"Message":"Login failed"}`);
47         }
48
49         // Error handling 2: If no record found for email/password entered (results is
         // empty array [])
50         else if (results.length == 0) {
51             res.status(404);
52             res.type("json");
53             res.send(`{"Message":"Login email or password incorrect"}`);
54         }
55
56         // Error handling 3: if email/password record is found but role is not admin
57         else if (results[0].role !== "admin") {
58             res.status(401);
59             res.type("json");
60             res.send(`{"Message":"Login failed as user is not an admin"}`);
61         }
62     });
63 }

```

```

62
63 // Login success as valid email/password is entered and is an admin
64 else {
65     res.status(200);
66     res.type("json");
67     // res.send(results);
68     res.send(`{"Message":"Login is successful!"}`);
69 }
70 }); //user.loginUser
71 });
72
73 // POST /genre - Add new genre record
74 app.post("/genre", function (req, res) {
75     var genre_name = req.body.genre_name;
76     var genre_description = req.body.genre_description;
77
78     genre.addGenre(genre_name, genre_description, function (err, results) {
79         // Error handling
80         if (err) {
81             res.status(500);
82             res.type("json");
83             res.send(`{"Message":"Add new genre failed!"}`);
84         }
85
86         // If no error
87         else {
88             res.status(200);
89             res.type("json");
90             res.send(`{"Message":"New Genre Added: ${genre_name}"}`);
91         }
92     });
93 });
94
95 // POST /movie - Add new movie record
96 app.post("/movie", function (req, res) {
97     var movie_name = req.body.movie_name;
98     var movie_description = req.body.movie_description;
99     var release_date = req.body.release_date;
100     var image_url = req.body.image_url;
101     var genre_id = req.body.genre_id;
102     var active = req.body.active;
103
104     movie.addMovie(
105         movie_name,
106         movie_description,
107         release_date,
108         image_url,
109         genre_id,
110         active,
111         function (err, results) {
112             // Error handling
113             if (err) {
114                 res.status(500);
115                 res.type("json");
116                 res.send(`{"Message":"Add new movie failed!"}`);
117             }
118
119             // If no error
120             else {
121                 res.status(200);
122                 res.type("json");
123                 res.send(`{"Message":"New Movie Added: ${movie_name}"}`);
124             }
125         }
126     );
127 });

```

```

124     }
125   );
126 });
127
128 // GET /genre - Retrieve all genre
129 app.get("/genre", function (req, res) {
130   genre.getGenre(function (err, results) {
131     // Error handling
132     if (err) {
133       res.status(500);
134       res.type("json");
135       res.send(`{"Message":"Get all genre failed"}`);
136     }
137
138     // If no error
139     else {
140       res.status(200);
141       res.type("json");
142       res.send(results);
143     }
144   });
145 });
146
147 // GET active movies or movies with substring & filter by genre
148 app.get("/movie", function (req, res) {
149   console.log(req.url);
150   console.log(req.method);
151   console.log(req.path);
152   console.log(req.query.active);
153   console.log(req.query.substr);
154   console.log(req.query.genreid);
155   var active = req.query.active; //get active query
156   var substr = "%" + req.query.substr + "%"; //get substr query
157   var genreid = req.query.genreid; //get genreid query
158
159   // GET /movie?active=Y - Retrieve all active screening movies
160   if (active) {
161     movie.getActiveMovie(active, function (err, results) {
162       // Error handling
163       if (err) {
164         res.status(500);
165         res.type("json");
166         res.send(`{"Message":"Get active screening movies failed"}`);
167       }
168
169       // If no error
170       else {
171         res.status(200);
172         res.type("json");
173         res.send(results);
174       }
175     }); //movie.getActiveMovie
176   }
177
178   // GET /movie?substr=<pat>&genreid=<num>
179   // Retrieve movies based on substring of movie name or genre id
180   else {
181     movie.getSortedMovie(substr, genreid, function (err, results) {
182       // Error handling
183       if (err) {
184         res.status(500);
185         res.type("json");

```

```
186         res.send(`{"Message":"Get sorted list of movies failed"}`);
187     }
188     // If no error
189     else {
190         res.status(200);
191         res.type("json");
192         res.send(results);
193     }
194 });
195 } //movie.getSortedMovie
196 });
197
198 module.exports = app;
199
```