

File: ./src/database.js

```
var sqlite3 = require('sqlite3').verbose()

let db = new sqlite3.Database('hk_complete.db', (err) => {
  if (err) {
    console.error(err.message);
    throw err
  }
  console.log('Connected to the H+K database.');
```

```
});

//create table for teammembers
var sql = 'CREATE TABLE tm (team_id TEXT, memberid TEXT)';
db.run(sql, (err) => {
  if (err) {
    console.log('Teammember Table already created');
```

```
  } else {
    console.log('Teammember Table created');
```

```
    var insert = 'INSERT INTO tm (team_id, memberid) VALUES (?, ?)';
1   db.run(insert, ["1", "1"])
2
3   db.run(insert, ["1", "2"])
4
5   db.run(insert, ["2", "1"])
6
7   db.run(insert, ["1", "3"])
8
9   db.run(insert, ["2", "3"])
10
11 }
12 }}
13
14 }}
15
16
17 // create table 'project'
18
19 sql='CREATE TABLE project (project_id TEXT, name TEXT, startdate DATE)';
20
21 db.run(sql, (err) => {
22
23   if (err) {
24
25     console.log('Project Table already created.');
```

```
  }else{
27
28     console.log('Project Table created.');
```

```
    insert = 'INSERT INTO project (project_id, name, startdate) VALUES (?, ?, ?)';
31
32     db.run(insert, ["1", "Project 1", '2020-12-10'])
33
34     db.run(insert, ["2", "Project 2", '2020-12-10'])
35
36     db.run(insert, ["3", "Project 3", '2020-12-10'])
37
38   }
39
40 }}
41
42 }}
43
44
45 // create table 'user'
46
47 sql='CREATE TABLE user (user_id INT NOT NULL UNIQUE, username TEXT, password TEXT, usertype TEXT, email TEXT)';
48
49 db.run(sql, (err) => {
50
51   if (err) {
52
53     console.log('User Table already created.');
```

```
  }
55
56   else {
57
58     //creating default users
59
60     console.log('User Table created.');
```

```
    var insert = 'INSERT INTO user (user_id, username, password, usertype) VALUES(?, ?, ?, ?)';
64
65     db.run(insert, [00001, "manager", "manager", "Manager"])
66
67
```

```

68     db.run(insert, [00002, "client", "client", "Client"])
69
70     db.run(insert, [00003, "employee", "employee", "Employee"])
71 }
72 }
});

```

```
module.exports = db
```

```
//TODO
```

```
/*
add
```

```
, projects ARRAY
```

```
to the User Table
*/
```

```
/*
add a KnowledgeTable
*/
```

File: ./src/hk_complete.db

[binary]

File: ./src/server.js

```

var express = require("express")
var cors = require('cors');
var bodyParser = require("body-parser");
var db = require("./database.js")

var app = express()

app.use(cors());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

var HTTP_PORT = 8085

// Start server

app.listen(HTTP_PORT, () => {
    console.log("Server running on port %PORT%".replace("%PORT%", HTTP_PORT))
});

// Root endpoint
app.get("/", (req, res, next) => {
    res.json({ "message": "Server is running on port %PORT%".replace("%PORT%", HTTP_PORT) })
});

```

```

//check login
app.post("/users/login/", (req, res) => {
  var errors = []
  if (errors.length) {
    res.status(400).json({ "error": errors.join(",") });
    return;
  }
  sql = 'SELECT * FROM user WHERE username = ? AND password = ?';
  params = [req.body.username, req.body.password]
  db.get(sql, params, (err, row) => {
    if (err) {
      res.status(400).json({ "error": err.message })
      return;
    }
    return row
      ? res.json({
          "message": "success",
          "data": row.usertype,
          "user_id": row.user_id
        })
      : res.json({
          "message": "invalid"
        })
  });
});
1
2
3
4
5
6 //get projects for id
7
8 app.post("/userproject/", (req, res, next) => {
9
10   var errors = []
11
12   var data = {
13
14     id: req.body.team_id
15
16   }
17
18   if (errors.length) {
19
20     res.status(400).json({ "error": errors.join(",") });
21
22     return;
23
24   }
25
26   sql = 'SELECT * FROM project WHERE project_id = ?';
27
28   params = [parseInt(data.id)]
29
30   db.get(sql, params, (err, result) => {
31
32     if(err) {
33
34       res.status(400).json({ "error": err.message })
35
36       return;
37
38     }
39
40     return result
41
42     ? res.json({
43
44       "message": "success",
45
46       "data": result
47
48     })
49
50     : res.json({
51
52       "message": "invalid"
53
54     })
55

```

```

55     });
56
57 });
58
59
60
61 // Create a new project
62
63 app.post("/projects/", (req, res, next) => {
64
65     var data = {
66
67         project_id: req.body.project_id,
68
69         name: req.body.name,
70
71         startdate: req.body.startdate,
72
73         projectteam: req.body.projectteam,
74
75         user_id: req.body.user_id
76
77     }
78
79     sql = 'INSERT INTO project (project_id, name, startdate) VALUES (?, ?, ?)';
80
81     var params = [data.project_id, data.name, data.startdate]
82
83     db.run(sql, params, function (err, result) {
84
85         if (err) {
86
87             res.status(402).json({ "error": err.message })
88
89             return;
90
91         }
92
93         res.json({
94
95             "message": "success",
96
97             "data": data
98
99         })
100
101     });
102
103 });
104
105
106
107 app.post("/teammember/", (req, res, next) => {
108
109     var data = {
110
111         project_id: req.body.project_id,
112
113         name: req.body.name,
114
115         startdate: req.body.startdate,
116
117         projectteam: req.body.projectteam,
118
119         user_id: req.body.user_id
120
121     }
122
123
124
125     sql = 'INSERT INTO tm (team_id, memberid) VALUES (?, ?)';
126
127     params = [data.projectteam.toString(), data.user_id.toString()]
128
129     db.run(sql, params, function (err, result) {
130
131         if (err) {
132
133             res.status(403).json({ "error": err.message })
134
135             return;
136
137         }
138
139         res.json({
140
141             "message": "success",
142
143             "data": data
144
145         })
146
147     })
148
149 });
150
151
152
153
154
155
156
157 //create new user
158
159 app.post("/users/", (req, res, next) => {

```

```

160
161     var data = {
162         user_id: req.body.user_id,
163         username: req.body.username,
164         password: req.body.password,
165         usertype: req.body.usertype
166     }
167
168     sql = 'INSERT INTO user (user_id, username, password, usertype) VALUES (?, ?, ?, ?)';
169
170     var params = [data.user_id, data.username, data.password, data.usertype]
171
172     db.run(sql, params, function (err, result) {
173         if (err) {
174             res.status(402).json({ "error": err.message })
175             return;
176         }
177         res.json({
178             "message": "success",
179             "data": data
180         })
181     });
182
183     //create new knowledge
184
185
186     // get teammembership
187     app.post("/tms/", (req, res, next) => {
188         let sql = 'SELECT team_id FROM tm WHERE memberid = ?';
189         params = [req.body.user_id]
190         db.all(sql, params, (err, rows) => {
191             if (err) {
192                 res.status(400).json({ "error": err.message });
193                 return;
194             }
195             res.json({
196                 "message": "success",
197                 "data": rows,
198                 "user": req.body.user_id
199             })
200         });
201     });
202
203     // Default response
204     app.use(function (req, res) {
205         res.status(404);
206     });
207
208     /*
209     // list all users DANGER
210     app.get("/users", (req, res, next) => {

```

```
    sql = `SELECT * FROM user`;
    var params = []
    db.all(sql, params, (err, rows) => {
      if (err) {
        res.status(400).json({ "error": err.message });
        return;
      }
      res.json({
        "message": "success",
        "data": rows
      })
    });
  });

  */
```

File: ./src/package-lock.json

1 [binary]

File: ./src/package.json

1 [binary]