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 (?, ?)';
    db.run(insert, ["1", "1"])
    db.run(insert, ["1", "2"])
    db.run(insert, ["2", "1"])
    db.run(insert, ["1", "3"])
    db.run(insert, ["2", "3"])
  }
})

// create table 'project'
sql='CREATE TABLE project (project_id TEXT, name TEXT, startdate DATE)';
db.run(sql, (err) => {
  if (err) {
    console.log('Project Table already created.');
  }else{
    console.log('Project Table created.');
    insert = 'INSERT INTO project (project_id, name, startdate) VALUES (?, ?, ?)';
    db.run(insert, ["1", "Project 1", '2020-12-10'])
    db.run(insert, ["2", "Project 2", '2020-12-10'])
    db.run(insert, ["3", "Project 3", '2020-12-10'])
  }
});

// create table 'user'
sql='CREATE TABLE user (user_id INT NOT NULL UNIQUE, username TEXT, password TEXT, usertype TEXT, email TEXT)';
db.run(sql, (err) => {
  if (err) {
    console.log('User Table already created.');
    }
    else {
      //creating default users
      console.log('User Table created.');
      var insert = 'INSERT INTO user (user_id, username, password, usertype) VALUES(?, ?, ?, ?)';
      db.run(insert, [00001, "manager", "manager", "Manager"])
      db.run(insert, [00002, "client", "client", "Client"])
      db.run(insert, [00003, "employee", "employee", "Employee"])
    }
});

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"
            })

    });
});


//get projects for id
app.post("/userproject/", (req, res, next) => {
    var errors = []
    var data = {
        id: req.body.team_id
    }
    if (errors.length) {
        res.status(400).json({ "error": errors.join(",") });
        return;
    }
    sql = 'SELECT * FROM project WHERE project_id = ?';
    params = [parseInt(data.id)]
    db.get(sql, params, (err, result) => {
        if(err) {
            res.status(400).json({ "error": err.message })
            return;
        }
        return result
        ? res.json({
            "message": "success",
            "data": result
        })
        : res.json({
            "message": "invalid"
        })
    });
    });

});



// Create a new project

app.post("/projects/", (req, res, next) => {

    var data = {

        project_id: req.body.project_id,

        name: req.body.name,

        startdate: req.body.startdate,

        projectteam: req.body.projectteam,

        user_id: req.body.user_id

    }

    sql = 'INSERT INTO project (project_id, name, startdate) VALUES (?, ?, ?)';

    var params = [data.project_id, data.name, data.startdate]

    db.run(sql, params, function (err, result) {

        if (err) {

            res.status(402).json({ "error": err.message })

            return;

        }

        res.json({

            "message": "success",

            "data": data

        })

    });

});



app.post("/teammember/", (req, res, next) => {

    var data = {

        project_id: req.body.project_id,

        name: req.body.name,

        startdate: req.body.startdate,

        projectteam: req.body.projectteam,

        user_id: req.body.user_id

    }



sql = 'INSERT INTO tm (team_id, memberid) VALUES (?, ?)';

params = [data.projectteam.toString(), data.user_id.toString()]

db.run(sql, params, function (err, result) {

    if (err) {

        res.status(403).json({"error": err.message})

        return;

    }

    res.json({

        "message": "success",

        "data": data

    })

})

});







//create new user

app.post("/users/", (req, res, next) => {
    var data = {
        user_id: req.body.user_id,
        username: req.body.username,
        password: req.body.password,
        usertype: req.body.usertype
    }
    sql = 'INSERT INTO user (user_id, username, password, usertype) VALUES (?, ?, ?, ?)';
    var params = [data.user_id, data.username, data.password, data.usertype]
    db.run(sql, params, function (err, result) {
        if (err) {
            res.status(402).json({ "error": err.message })
            return;
        }
        res.json({
            "message": "success",
            "data": data
        })
    });
});
//create new knowledge


// get teammembership
app.post("/tms/", (req, res, next) => {
    let sql = 'SELECT team_id FROM tm WHERE memberid = ?';
    params = [req.body.user_id]
    db.all(sql, params, (err, rows) => {
        if (err) {
            res.status(400).json({ "error": err.message });
            return;
        }
        res.json({
            "message": "success",
            "data": rows,
            "user": req.body.user_id
        })
    });
});

// Default response
app.use(function (req, res) {
    res.status(404);
});









/*
// list all users DANGER
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

```
[binary]
```

File: ./src/package.json

```
[binary]
```