Part 4: Server Design Document
Members: Matt G. Matt H. Micheal J.

Notes: Endpoints work as follows: Userinfo has permissions and data stored -> Projects to view all projects -> {project} is specific project id to view and all tasks - >/userinfo/{task}/ edit or read changes depending on userinfo permissions

Route	Method	Protected/Public	Description
/login	GET/POST	Public	Log-in for session/Affects the rest of the methods by changing server state
/register	GET/POST	Public	Register a user in system
/users	GET/PUT/DEL	Protected	Admin view all users/Manage Users
/userinfo	GET/PUT/DEL	Protected	Base user session/Edit own user info(Name/Email/Pass)
/userinfo/projects	GET/Put/Del/POST	Protected	Call request for all projects assigned to user Manage all projects (Admin view)
/userinfo/{project}/tasks	GET/POST	Protected	View Tasks under a project/Edit tasks, if permissions
/userinfo/{task}/edit	GET/POST/DELETE/PUT	Protected	User editing/adding based on user permission in specific task
/userinfo/{task}/read	GET	Protected	View only of task not assigned to user

/login

Responsibilities: Get Login page, and Post login information. This endpoint does the authentication for users, and creates the session that will change their view of projects. Redirects to /register, /userinfo/projects, or /users depending on authentication.

Tooling: Express,Bcrypt, shortid, passport

/register

Responsibilities: Register a new account. Redirects to /login.

Tooling: Express,Bcrypt, shortid, passport

/users

Responsibilities: Redirected from login, when ADMIN is auth. Views all Users, and who is assigned what.on the /edit, allows for addition/removal of user accounts, projects, and tasks. Redirects to /login

Tooling: Express

/userinfo

Responsibilities: Userinfo stored in session. Judges permissions of all future endpoints. On its own, is the editing for userinfo (Rather than calling a “Settings”). Redirects to /login and /userinfo/projects.

Tooling: Express

/userinfo/projects

Responsibilities: Displays all projects currently in the database. Prioritize projects based on which user is assigned. Option to add a new project. Redirects to /login, /userinfo/{project}/tasks, /userinfo, and /userinfo/{project}.

Tooling: Express

/userinfo/{project}

Resonsibilities: Manage tasks/users for each project. Available to those with manager/ADMIN privileges

Tooling: Express

/userinfo/{project}/tasks

Responsibilities: Displays all tasks for a given project. Option for manager to add new tasks. Checks user permission to determine whether they can edit or read tasks. Redirects to /login, /userinfo/{task}/edit or /userinfo/{task}/read, and /userinfo/projects.

Tooling: Express

/userinfo/{task}/edit

Responsibilities: Displays and allows edits for a task within a project. Changes model data. Redirects to /login and /userinfo/{project}/tasks.

Tooling: Express

/userinfo/{task}/read

Responsibilities: Displays a task within a project. Redirects to /login and /userinfo/{project}/tasks.

Tooling: Express

Example UserInfo JSON -


```
{ "name": "First Last",
  "email": "first.last@example.com",
“Password”:”xxxxxxxxxxx”(Hashed password)
  "Projects": [
    {
      "Project_id": "xxxxxx”
    },
    {
      "Project_id": "xxxxx”
        }
      ]
}
```


Example ProjectInfo:


```
{“ProjectId”:”xxxxxxxxxxxx”,
“Tasks”:[
“Taskid”:”xxxxxxxx”,
“Taskid”:”xxxxxxxx”],
“Users”:[],
“Manager”:”xxxxxx”
}
```


Example Task:


```
{“TaskId”:”xxxxx”,
“DueDate”:”xxxxxxxx”,
“Deliverable”:file/.doc/link,
“Assinged_Users”:[],
“Manager”:user
}
```