

## CS5500 WeGroup Document

This factor evaluates the external project documentation. This includes project organization, requirements, features, functionality interfaces or APIs, design, communications plans, and test plans

### Project organization:

We organize the whole project in four folders. They are called backend, frontend, docs, and readme. They contain different files inside which play different roles in the entire project. Here are the basic introductions of all four folders.

#### backend:

Basically, this folder contains all the technical related files of backend design.

We have a subfolder called “models”, which contains all the data schemas we designed as the database.

We have a subfolder called “routes”, which contains all the API files to provide the functionality.

We have a subfolder called “middleware”. It contains the authentication related files.

We also have the server related files contained in this folder.

#### frontend:

Basically, this folder contains all the technical related files of frontend design.

We have a subfolder called “public”, which contains all the HTML entry files.

We have a subfolder called “styles”, which contains all the css file.

We have a subfolder called “src”, which contains all the course code, for example, the components and services.

We have a subfolder called “service”, which helps to connect the frontend with the backend.

#### docs:

Basically, this folder contains all the technical related files for everyone to read.

“The communication plan” is the file to record our meetings about the project design.

“CS5500 WeGroup Document” is the file which includes all the introductions to help the professor and teaching assistants evaluate our project.

#### readme:

This is the file we provide for other developers.

In this file, we have the instructions for other developers to install the servers, run the build, and test all the functions.

## Project requirements:

A full-stack web application to let users create groups and collaborate with each other to share messages(string, photos, files, emoji), calendar.

FrontEnd: React, Figma

BackEnd: Node.js, Postman for testing

Database: MongoDB

## Project features:

We have several features:

- Sign up/ Log in
- Post
- Search
- Team

## Project API:

User:

- createUser()
- updateUser()
- getUser()
- deleteUser()

Post:

- createPost()
- updatePost()
- getPostCreatedByUser()
- deletePost()
- getPostsbyTag()
- getPostsInteratedByUser()
- getTeamSize()

Team:

- createTeam()
- deleteTeam()
- getInterestedUserList()
- getTeam()

Tag:

- createTag()
- searchTag()
- getAllTags()
- getTagByTagId()

Comment:

- createComment()
- deleteComment()
- editComment()
- deleteComment()

## Project Design:

Frontend Design: [FronEnd Design](#)

Backend Design: [BackEnd Design](#)

## Project communication plans:

[communication plans](#)

## Project test plans:

We used Postman to test our backend API and log error message in the chrome inspect console to test frontend

1. create each component with empty function, render the components using chrome to check component render as expected
2. add fake data for components to render, using chrome to check component render data as expected
3. add functions and logics to buttons and links and using chrome to test components behave as expected. using console log for testing and debugging and using chrome inspect to see the console logs.

4. connect components to the backend, render the component using fetched data. Check in Chrome to verify the react application interacts with the backend as designed.

```
Line 67: PostService is defined but never used      no-unused-vars
> POST http://localhost:8080/post net::ERR_CONNECTION_RESET      PostService.js:27
> Uncaught (in promise) TypeError: Failed to fetch                create_post:1

undefined
(2) [{_, {}}]
  0: {posts: Array(0), _id: "5fcffe58c0c2f34fccf5e711", department: "Computer Science", courseNumber: 5500, year: 2020, ...}
  1: {posts: Array(0), _id: "5fcffe63c0c2f34fccf5e712", department: "Computer Science", courseNumber: 6140, year: 2020, ...}
    length: 2
  __proto__: Array(0)
```

```
TypeError: Cannot read property 'getAllTags' of undefined
(anonymous function)
src/components/PostForm.js:12

   9 |
  10 | const [allTags, setAllTags] = useState([]);
  11 | useEffect(()=>{
> 12 |   tagService.getAllTags().then(res => res.json()).then(res => {
    |   ^
  13 |     setAllTags(res.data);
  14 |     if(allTags.length > 0) {
  15 |       console.log("all tags:" + allTags[0]);
```

