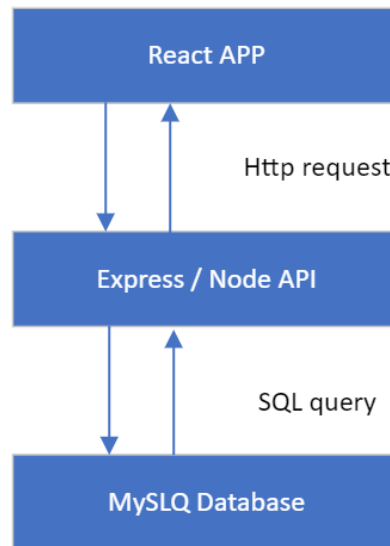


Technical Specification Sheet

This is a web application containing a front end, a back end, and a SQL database. the front end is based on React which is a JavaScript library for building user interfaces. The back end is based on Express, a minimal and flexible Node.js web application framework. The database is stored in MySQL, a relational database management system.



This is an explanation of all the APIs in this application:

Create a project

POST ⌵ `http://localhost:3002/project/create`

Params Authorization Headers (9) **Body** ● Pre-request Script

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary

```
1 {
2   ... "organizationName": "organizationName new 2",
3   ... "projectTitle": "projectTitle new 2",
4   ... "deliverables": "deliverables new 2",
5   ... "knowledgeSkills": "knowledgeSkills new 2",
6   ... "projectDescription": "projectDescription new 2"
7 }
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize **JSON** ⌵ ⌵

```
1 {
2   "error": false,
3   "message": "Project create successfully"
4 }
```

Check projects list

GET

▼

http://localhost:3002/project/list

ParamsAuthorizationHeaders (7)BodyPre-request ScriptTest


Query Params

	KEY	VALUE
--	-----	-------

BodyCookiesHeaders (10)Test Results

PrettyRawPreviewVisualizeJSON

▼



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

{
 "ID": 1,
 "OrganizationName": "organizationName22",
 "ProjectTitle": "projectTitle22",
 "Description": "projectDescription22",
 "Deliverables": "deliverables22",
 "KnowledgeSkill": "knowledgeSkills22"
},
{
 "ID": 2,
 "OrganizationName": "organizationName11",
 "ProjectTitle": "projectTitle11",
 "Description": "projectDescription11",
 "Deliverables": "deliverables11",
 "KnowledgeSkill": "knowledgeSkills11"
},
{
 "ID": 3,

Check the details of a project with Project ID

GET

▼

http://localhost:3002/project/detail/2

ParamsAuthorizationHeaders (7)BodyPre-request Scr


Query Params

	KEY	VAL
--	-----	-----

BodyCookiesHeaders (10)Test Results

PrettyRawPreviewVisualizeJSON

▼



1

2

3

4

5

6

7

8

{
 "ID": 2,
 "OrganizationName": "organizationName11",
 "ProjectTitle": "projectTitle11",
 "Description": "projectDescription11",
 "Deliverables": "deliverables11",
 "KnowledgeSkill": "knowledgeSkills11"
}

Upload a student

(studentID, familyName, givenName, title, email, gpa, perference, skills, teamName)

POST

http://localhost:3002/team/create

Params

Authorization

Headers (9)

Body

Pre-request Sc

none

form-data

x-www-form-urlencoded

raw

bi

```
1 {
2   "studentID": "12312315",
3   "familyName": "familyName new 2",
4   "givenName": "givenName new 2",
5   "title": "title new 2",
6   "email": "email new 2",
7   "gpa": "gpa new 2",
8   "perference": "perference new 2",
9   "kills": "skills new 2",
10  "teamName": "teamName new 2"
11 }
```

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "error": false,
3   "message": "Student create successfully"
4 }
```

Check the list for all the teams

GET

http://localhost:3002/team/list

Params

Authorization

Headers (7)

Body

Pr

Query Params

KEY

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 [
2   "teamName new 1",
3   "Team A",
4   "Team B"
5 ]
```

Check details of students in a team with Team Name


GET ⌵ http://localhost:3002/team/detail/Team B

Params Authorization Headers (7) Body Pre-request Script

Query Params

	KEY	VALUE
--	-----	-------

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ⌵ 

```
1 [
2   {
3     "StudentID": "52345678",
4     "FamilyName": "familyName5",
5     "GivenName": "givenName5",
6     "Title": "Ms",
7     "Email": "email5",
8     "Gpa": "gpa5",
9     "Perference": "perference5",
10    "Skills": "skills5",
11    "TeamName": "Team B",
12    "ProjectID": 2
13  },
14  {
15    "StudentID": "62345678",
16    "FamilyName": "familyName6",
17    "GivenName": "givenName6",
18    "Title": "Ms",
19    "Email": "email6",
```

Check teams without allocation


GET ⌵ http://localhost:3002/allocate/teams

Params Authorization Headers (7) Body Pre-request

Query Params

	KEY	VALUE
--	-----	-------

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ⌵ 


```
1 [
2   "Team A",
3   "Team B"
4 ]
```

Check projects without allocation

GET ⌵ http://localhost:3002/allocate/projects

Params Authorization Headers (7) Body Pre-request S

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ⌵ 

```
1 [
2   {
3     "ID": 1,
4     "ProjectTitle": "projectTitle22"
5   },
6   {
7     "ID": 2,
8     "ProjectTitle": "projectTitle11"
9   },
10  {
11    "ID": 3,
12    "ProjectTitle": "projectTitle33"
13  },
14  {
15    "ID": 4,
16    "ProjectTitle": "projectTitle0"
17  },
18  {
19    "ID": 5,
20    "ProjectTitle": "projectTitleX"
```

Upload the allocation (project - team)


POST ⌵ http://localhost:3002/allocate/allocate

Params Authorization Headers (9) Body ● Pre-request S

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ b

```
1 {
2   ... "projectId": 9,
3   ... "teamName": "Team C"
4 }
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ⌵ 

```
1 {
2   "error": false,
3   "message": "Student create successfully"
4 }
```

Check the list of all the allocations

GET

▼

http://localhost:3002/allocate/allocation

ParamsAuthorizationHeaders (7)BodyPre-request Script

BodyCookiesHeaders (10)Test Results

PrettyRawPreviewVisualizeJSON▼

1

[

2

{

3

"TeamName": "Tree Team",

4

"ProjectID": 6

5

},

6

{

7

"TeamName": "Banana Team",

8

"ProjectID": 2

9

},

10

{

11

"TeamName": "Apple Team",

12

"ProjectID": 12

13

},

14

{

15

"TeamName": "King Team",

16

"ProjectID": 1

17

}

18

]