

# Legendary Funicular

**Networking Revolutionised**

[GitHub - Legendary Funicular](#)

RA1811028010016 - Shreyas Raorane

RA1811028010017 - Hritwik Mukherjee

# Abstract

Social networking for Students is primarily limited to websites such as LinkedIn and college based contacts. This allows students to search for like-minded people only in a limited circle. In contrast, using our application would enable students to expand their boundaries and reach students from various parts of the world.

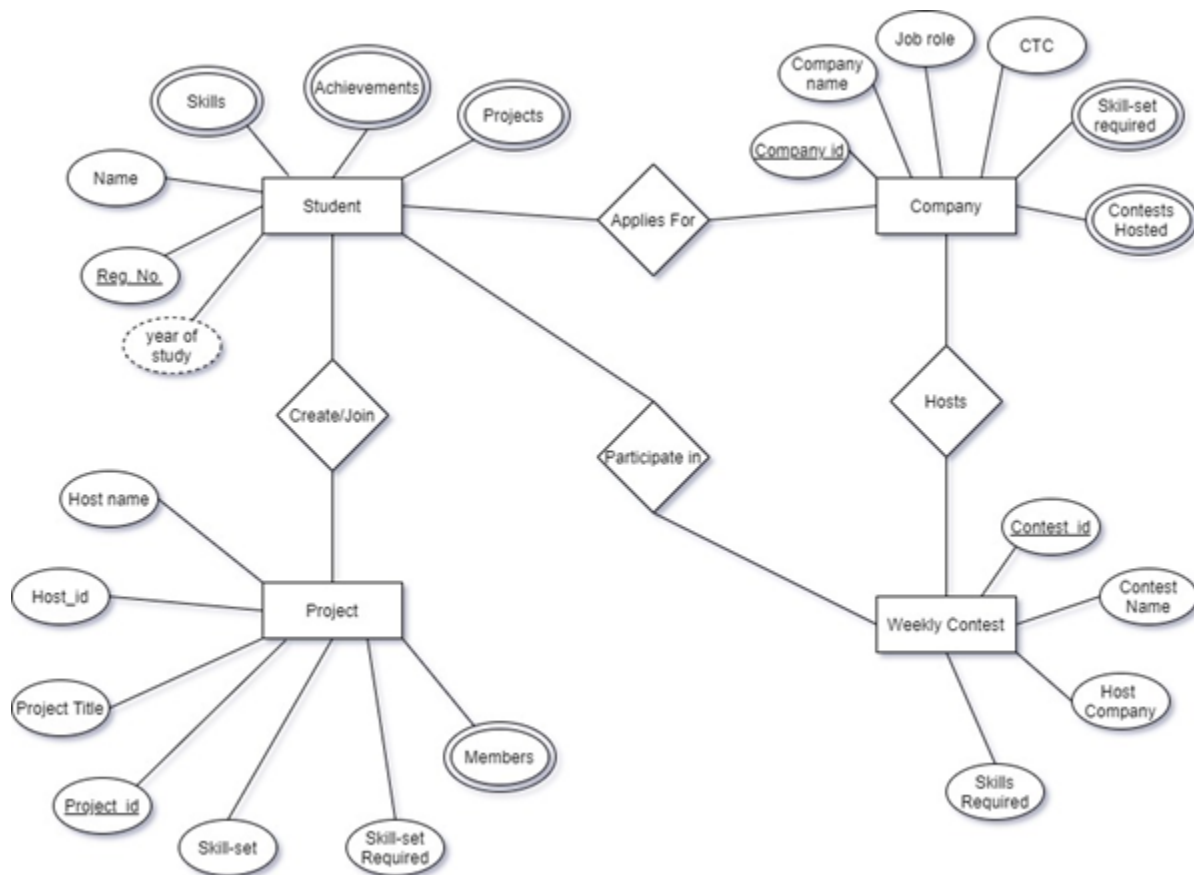
This is a social networking website designed for business professionals. Interaction on the platform is not restricted to only companies to students but collaborations between students is also encouraged.

It allows you to share work-related information with other users and keep an online list of professional contacts.

The project highlights the database management aspect of the aforementioned application.

# Introduction

The platform(*Legendary Funicular*) receives and processes data concerning students on the platform, companies registered on the platform, projects being collaborated upon and weekly contests that are being conducted. Hence the database is composed of four tables namely **Student**, **Company**, **Project**, **Weekly Contest**.



## Student Table

The Student table holds data such as the student's registration number, skills or their achievements. The platform therefore maintains a record of registered students so they can collaborate on projects and preferred candidates can be contacted by companies.

```
mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Reg_no         | varchar(255)  | NO   | PRI | NULL    |       |
| Name           | varchar(255)  | YES  |     | NULL    |       |
| YOS            | int           | YES  |     | NULL    |       |
| Skills         | varchar(255)  | YES  |     | NULL    |       |
| Achievements  | varchar(255)  | YES  |     | NULL    |       |
| Projects       | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.17 sec)
```

## Company Table

The Company table holds the data of companies and institutions registered on the platform. The data can be viewed by students who may want to apply.

```
mysql> desc company;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Company_ID     | varchar(255)  | NO   | PRI | NULL    |       |
| Company_name   | varchar(255)  | YES  |     | NULL    |       |
| CTC            | int           | YES  |     | NULL    |       |
| Job_Role       | varchar(255)  | YES  |     | NULL    |       |
| Skill_set      | varchar(255)  | YES  |     | NULL    |       |
| Contests       | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## Project Table

The project table maintains information about Projects being conducted on the platform. Students can be recruited for projects and can also apply for projects.

```
mysql> desc projects;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Project_id     | varchar(255)  | NO   | PRI | NULL    |       |
| Project_title  | varchar(255)  | YES  |     | NULL    |       |
| Host_id        | varchar(255)  | YES  |     | NULL    |       |
| Host_name      | varchar(255)  | YES  |     | NULL    |       |
| Skill_set      | varchar(255)  | YES  |     | NULL    |       |
| Skill_set_required | varchar(255) | YES  |     | NULL    |       |
| Members        | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

## Weekly Contest table

This table maintains a record of the contests that have been held and are being held on the platform.

```
mysql> desc weekly_contests;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Contest_id     | varchar(255)  | NO   | PRI | NULL    |       |
| Contest_name    | varchar(255)  | YES  |     | NULL    |       |
| Host_Company    | varchar(255)  | YES  |     | NULL    |       |
| Skills_Required | varchar(255)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

# Proposed Work

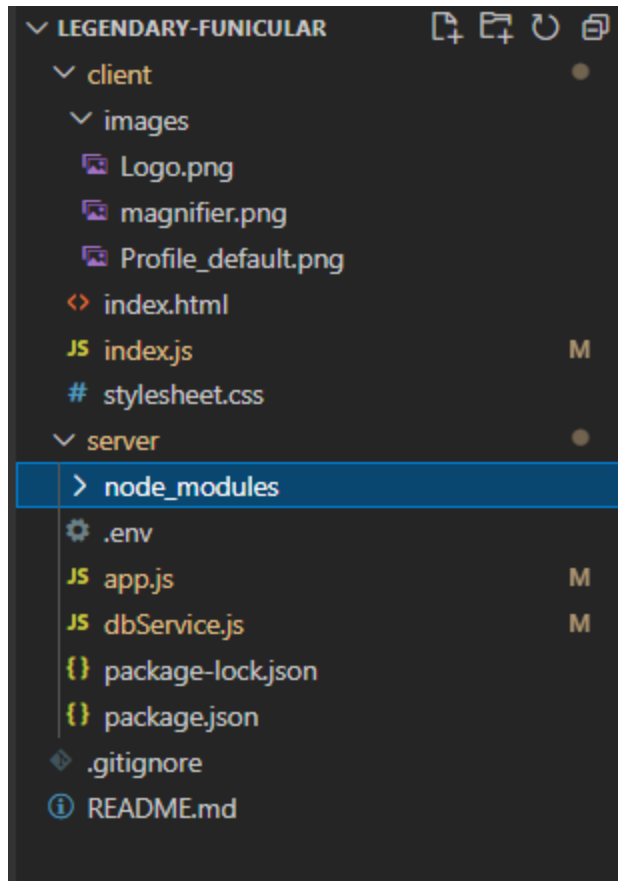
Legendary Funicular aims to create a better networking environment for students and academics to network and collaborate on various endeavours. It also provides institutions and companies with a platform where they can hire and nurture talent.

In this day and age communication and collaboration is essential to the growth of an individual and can be deciding factors in shaping one's future. Hence we are providing a platform to the individuals who want to make it further than anyone else and ones who hold their ambition in high regard.

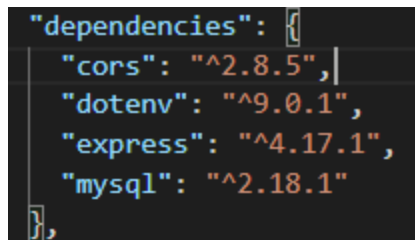
## Implementation

The databases for legendary funicular have been created in MySQL and the web portal has been developed in JavaScript.

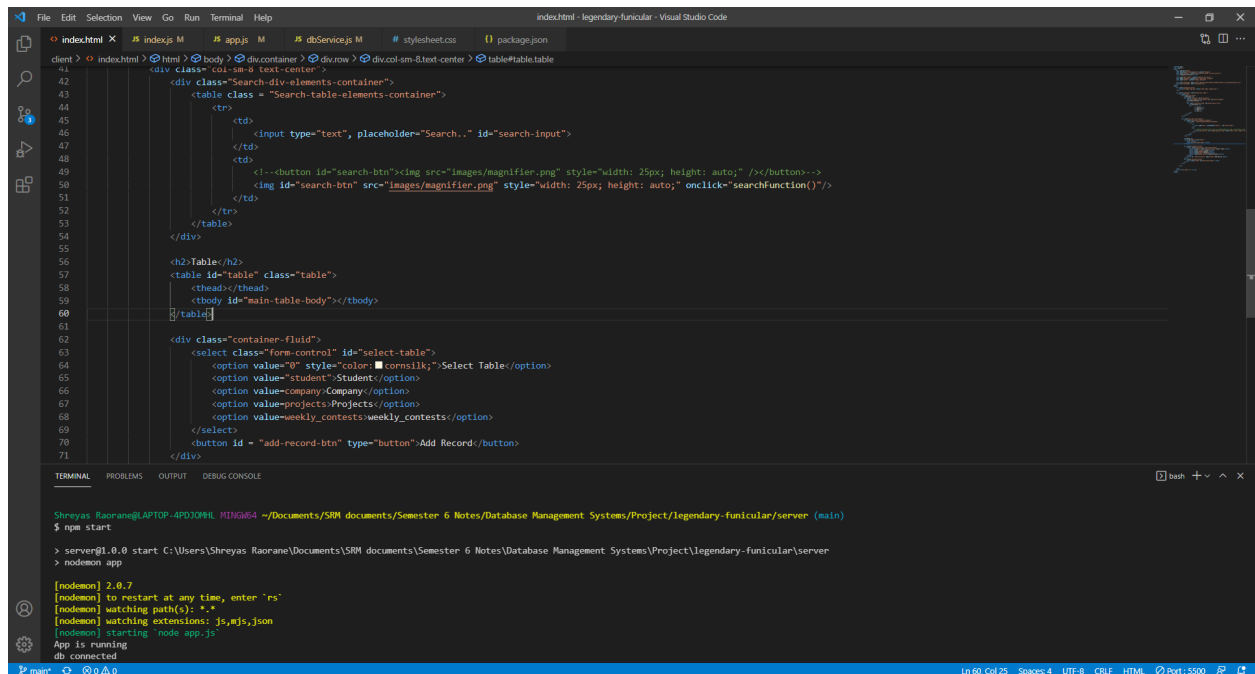




File Structure



Node Dependencies



Index.html: Landing page for the Website





The screenshot shows a Visual Studio Code editor with a dark theme. The editor has several tabs open: 'index.html', 'index.js', 'app.js', 'dbService.js', 'stylesheet.css', and 'package.json'. The 'dbService.js' tab is active, displaying the following code:

```
server > dbService.js > DbService > insertCompanyData > New_company_table_val > <function>
1 const mysql = require('mysql');
2 const dotenv = require('dotenv');
3 const { request, response } = require('express');
4 const e = require('express');
5 let instance = null;
6 dotenv.config();
7
8 const connection = mysql.createConnection({
9   host: process.env.HOST,
10  user: process.env.USER,
11  password: process.env.PASSWORD,
12  database: process.env.DATABASE,
13  port: process.env.DB_PORT
14 });
15
16 connection.connect((err) => {
17   if (err) {
18     console.log(err.message);
19   }
20   console.log('db ' + connection.state);
21 });
22
23 class DbService {
24   static getServiceInstance() {
25     return instance ? instance : new DbService();
26   }
27
28   async getTableData(tname) {
29     try {
30       console.log(tname);
31       const response = await new Promise((resolve, reject) => {
```

Below the editor is a terminal window. It shows the command prompt for 'Shreyas Raorane@LAPTOP-4P0309H1 MINGW64' and the execution of 'npm start'. The output shows the server starting successfully and the application running on port 3000. The terminal also shows the command 'nodemon app' and the output of 'nodemon' starting the application.

**dbService.js:** Used in app.js, contains sql commands and corresponding functions to execute and return the result

To view the code in greater detail visit: <https://github.com/Shreyas0812/legendary-funicular>

# Output

Legendary Funicular provides users with the capability to view records of students, companies, contests and projects which have been registered on the platform.

## Students Table

The screenshot shows a web browser window with the URL `127.0.0.1:5500/client/index.html`. The page has a light blue header with the "FUNICULAR" logo. The main content area is divided into two sections: a "Profile" section on the left and a "Table" section on the right.

**Profile Section:**

- Header: "Profile"
- Search bar: "Search..." with a magnifying glass icon.
- Profile picture: A circular icon with a network diagram.
- Name: "Shreyas Raorane"
- Top Skills:
  - Python
  - Fortran
  - HTML
  - css

**Table Section:**

Reg_no	Name	YOS	Skills	Achievements	Projects	Delete
S1801	Ramesh	3	C,C++,C#,Python	Winner-WC2101	Crime rate Predictions using the K-means algorithm	Delete
S1802	Suresh	3	C	RunnerUp-WC2101		Delete
S1803	Naresh	3	C,C++,C#	Winner-Painting Competition	Detection of Brain Tumor with the help of image segmentation	Delete
S1901	Gregory Rasputin	2	C,C++	RunnerUp-Poetry		Delete

## Company Table


legendary-funicular

127.0.0.1:5500/client/index.html

Apps Gmail YouTube Maps Translate

FUNICULAR

Profile



Shreyas Raorane

Top Skills

- Python
- Fortran
- HTML
- CSS

Search..

Table

Company_id	Company_name	Job_Role	Skill_set	Contests	Delete
C01	Shell	Intern	SQL, Javascript		Delete
C02	Microsoft	Intern	Cyber Security	WC2103	Delete
C03	GameLoft	Game Developer	C++, C#	WC2104	Delete
C04	Paypal	Software Engineer	C++, C#, Python		Delete
C05	Reliance	Software Engineer	C++, C	WC2105	Delete
C06	Cisco	Network Analyst	Network Analyst Computer Networking	WC2106	Delete

Company

Add Record

Add New:

## Projects Table


legendary-funicular

127.0.0.1:5500/client/index.html

Apps Gmail YouTube Maps Translate

FUNICULAR

Profile



Shreyas Raorane

Top Skills

- Python
- Fortran
- HTML
- CSS

Search..

Table

Project_id	Project_title	Host_id	Host_name	Skill_set	Skill_set_required	Members	Delete
P01	Crime rate Predictions using the K-means algorithm	S1801	Ramesh	C, C++, C#, Python	Web Development	S1801	Delete
P02	Detection of brain tumor with the help of image segmentation	S1803	Naresh	C, C++, C#	Medicine	S1803	Delete
P03	Rock, paper, scissor game using python	S2001	Bruno Mars	C, C++, C#	Python	S2001, S2002, S2003	Delete
P04	Estimating oil prices for the decade	C01	Shell		AI, ML		Delete
P05	Payment Portal using SQL	C04	Paypal		SQL		Delete
P06	Website	C04	Paypal		Javascript		Delete

## Contests Table

The screenshot shows the Funicular web application interface. On the left is a user profile for Shreyas Raorane with a list of top skills: Python, Fortran, HTML, and CSS. The main section is titled 'Table' and contains a table of contests. A search bar at the top of the table is empty. Below the table is a dropdown menu set to 'weekly\_contests' and an 'Add Record' button. At the bottom, there is an 'Add New:' section with input fields for Contest\_id, Contest\_name, Host\_Company, and Skills\_Required.

Contest_id	Contest_name	Host_Company	Skills_Required	Delete
WC2101	CodeAway01	Microsoft	Problem Solving, C, C++	Delete
WC2102	Code for a better future	Reliance	Problem Solving, C, C++, Python	Delete
WC2103	CodeAway02	Microsoft	Problem Solving, C, C++	Delete
WC2104	GameJam1.0	Gameloft	Unity, UE4	Delete
WC2105	Code for a better future	Reliance	Problem Solving, C, C++, Python	Delete
WC2106	CaptureTheFlag	Cisco	Problem Solving, Linux Environment	Delete

## Search Functionality

Users can also search through records on the platform with the help of a search bar provided on the platform. Users can provide keywords or terms to search against the database.

This screenshot shows the same Funicular web application interface, but with the search bar in the 'Table' section populated with the text 'pyt'. The table below now displays search results with columns: Reg\_no, Name, YOS, Skills, Achievements, Projects, and Delete. Below the table, the dropdown menu is set to 'Student' and the 'Add Record' button is visible. The 'Add New:' section at the bottom has input fields for Reg\_no, Name, YOS, Skills, Achievements, and Projects.

Reg_no	Name	YOS	Skills	Achievements	Projects	Delete
S1801	Ramash	3	C,C++,C#,Python	Winner - WC2101	Crime rate Predictions using the K-means algorithm	Delete
S1903	David Beckham	2	C,C++,C#,Python	Winner - Cricket Tournament		Delete
S2001	Bruno Mars	1	C,C++,C#	Winner - Singing Competition	Rock, paper, scissor game using python	Delete
S2002	Ben Affleck	1	C,C++,C#,Python		Rock, paper, scissor game using python	Delete
S2003	Mark Ruffalo	1	C,C++,C#,Python		Rock, paper, scissor game using python	Delete


legendary-funicular x +

127.0.0.1:5500/client/index.html ☆ S

Apps Gmail YouTube Maps Translate

**FUNICULAR**

## Profile



**Shreyas Raorane**

**Top Skills**

- Python
- Fortran
- HTML
- CSS

avi 🔍

## Table

Reg_no	Name	YOS	Skills	Achievements	Projects	Delete
S1903	David Beckham	2	C,C++,C#,Python	Winner - Cricket Tournament		Delete

Student ▼ Add Record

Add New:

Reg_no	Name	YOS	Skills	Achievements	Projects
--------	------	-----	--------	--------------	----------

Deletion of records is also possible.

# References:

<https://www.youtube.com/watch?v=vrj9AohVhPA>

[https://www.youtube.com/watch?v=Uvy\\_BlgwfLI](https://www.youtube.com/watch?v=Uvy_BlgwfLI)

<https://www.youtube.com/watch?v=eljbSH3lmb8&t=227s>

<https://www.tutorialspoint.com/nodejs/index.htm>

<https://www.w3schools.com/bootstrap/>

<https://www.w3schools.com/nodejs/>

[https://www.w3schools.com/nodejs/nodejs\\_mysql.asp](https://www.w3schools.com/nodejs/nodejs_mysql.asp)

<https://www.javatpoint.com/expressjs-tutorial>