

# Acknowledgment

We wish to acknowledge with gratitude, the insightful guidance given by our module leader Dr. Rasika Ranaweerage. You have been our guidance mentor. We have been extremely lucky to have a lecturer who cared so much about student work, and who responded to our questions so promptly regarding. Not only module contents whenever he taught many extra trends, but technologies regarding our module since the second year and it also increase our knowledge.

# Table of Contents

# Table of figures

# Project identification

## Introduction

University to Industry is a platform that allows the industry personals to directly recruit university students. The platform allows a registered university student to maintain his profile and curriculum vitae which are made visible for the registered industry personals. The industry professionals can find out the students with qualifications and skills that they are looking for using the platform.

This project is mainly about the development of the API for this system. The REST API has been developed for the CRUD operations of the database that is being used for this system. The system is also consisted of a web application as the front end that uses the API to communicate with the database.

## Project Goals

The main goal of this project is to develop a RESTful API web application and complete our project according to the given assignment criteria.

Learn something new and improve our technical skills and knowledge were our second goal as a team.

## Project Objectives

* Carefully analyses the scenario and deliver a good effective solution.
* Figure out new developing tools and technologies which can fulfill the project requirements.
* Design and development of a functional web application and a mobile application.
* The source code should have been tested properly by using test data.
* The design clearly illustrated within the website documentation and clearly evident in the architecture.
* Proper project documentation.

# Planning

Here, we identified and defined the purpose and project plan for the system development. There is the main task we have completed in this stage.

That is Requirement Gathering. Requirements gathering is an essential part of any project and project management. Understanding fully what a project will deliver is critical to its success.

Requirement gathering was mainly done by analyzing project criteria and observing the given scenario.

# Analyze

## Functionalities

From the IPT program manager(Admin) aspect,

* IPT program manager can access the websites admi panel by using username and password.
* IPT program manager can see details about any registered member.
* According to the given details IPT program manager can approve or reject any membership.
* IPT program manager can modify their account details and logout from the account.

From the student aspect,

* Any student can sign up for the website by giving required details.
* They can sign in using username and password.
* Student can edit their profile details and share their qualifications.

From the IT expert aspect,

* Any IT expert can sign up for the website by giving required details.
* They can sign in using username and password.
* Approved experts can filter by the category to find appropriate students and communicate with them.
* They can also can filter by the category and see the previous selected students and their contact details.

## Requirement Analysis

### Functional Requirement

1. Students can register themselves with details including a student ID, name, profession, email, affiliated university, password, etc.
2. Experts can register themselves with details include national ID, name, profession, email, affiliated company, password, etc.
3. Students or experts can update their profiles.
4. Anyone who logged in can access member details when the id is known.
5. An expert should be able to filter the students by category.
6. The IPT manager can remove invalid members, students or experts

### Non-Functional Requirements

* 1. Availability- The system is available for accessing anytime with the maximum number of users
  2. Reliability- The system is error-free and work according to the specifications mentioned.
  3. Security- A third party cannot access the system without any authorization from the administration.
  4. Accuracy- The system depends on real time information and the system provides real time updated data to the user. If any modification happened that will be updated in the system instantly.
  5. Maintainability- The system can be easily maintained.

# Designing

## Architecture Diagram

A screenshot of a cell phone

Description automatically generated

Figure I -Architecture Diagram

## Use Case Diagram

A picture containing text, map

Description automatically generated

Figure II- Use Case Diagram

## Database Design

### Extended Entity Relationship Diagram

A picture containing text, map

Description automatically generated

Figure III- EER

### Relational Mapping

A screenshot of a cell phone

Description automatically generated

Figure IV - Relational Mapping

### Normalized Tables

A screenshot of a cell phone

Description automatically generated

Figure V - Normalized Tables

### Database Diagram

A screenshot of a cell phone

Description automatically generated

Figure VI - Database Diagram

# Implementation

## Web App Implementation

### Features of Web Application

### Web Application Workflow

## Mobile App Implementation

### Features of Mobile Application

### Mobile Application Workflow

## API Implementation

### Overview

Apart from node modules in the backend **API** folder is heart of this application. Because it contains Restful Web API.



Figure VII- Project Structure

In server.js file contains configurations of server. It has whole responsible of routing and same time working as server.



Figure VIII- server.js

### API Project Dependencies

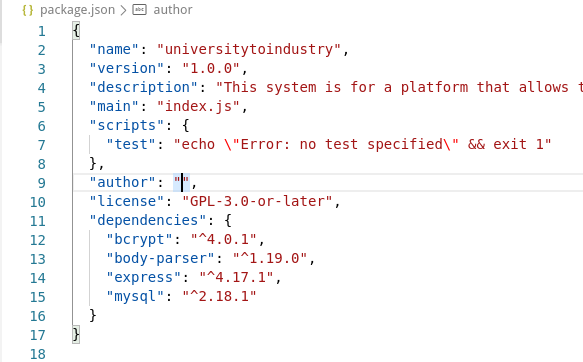


Figure IX- package.json

### Model

First, we designed EER and summarize table meta details as in normalized table section.

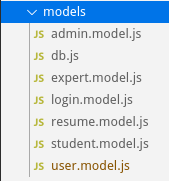


Figure X- Models

Also, in the backend folder it has db.js file which is store connection string to the database.

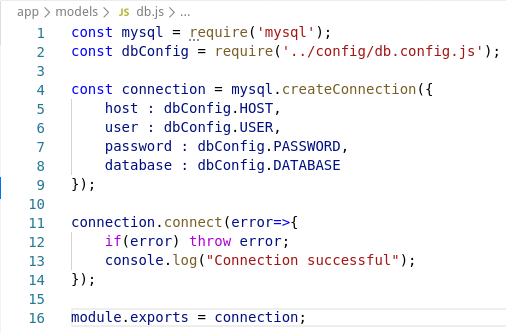


Figure XI – db.js

For an example we added a part of the User model.



Figure XII - A part of the user.model.js

### Controller

After implementing models then we implemented controllers for models. In controllers we import models and ad methods which are GET, POST, PUT, DELETE HTTP methods.

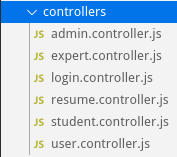


Figure XIII – Controllers

For example we added a part of the user controller that contains the controller of just record creation.



Figure XIV- a part of the user.controller.js

### Routes

Routes are defining the URI of particular resources along with certain HTTP methods.

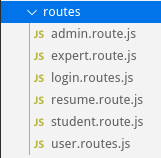


Figure XV-Routes



Figure XVI -admin.routes.js



Figure XVII - expert.routes.js

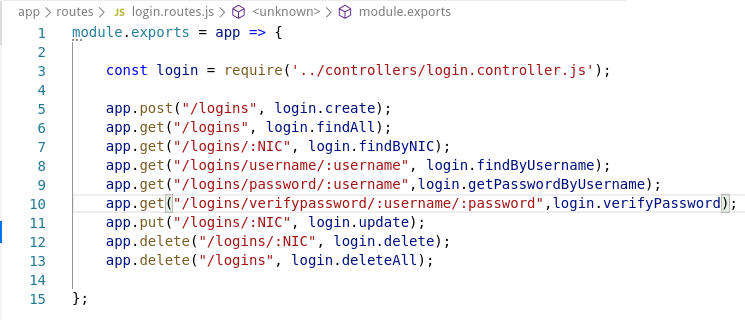


Figure XVIII - login.routes.js

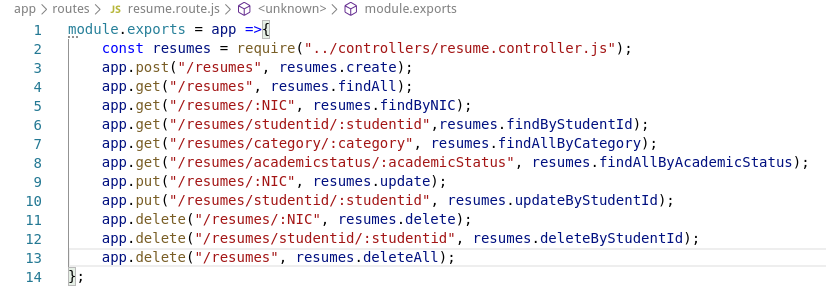


Figure XIX - resume.route.js

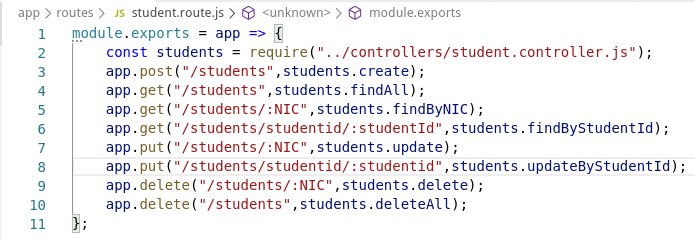


Figure XX - student.route.js

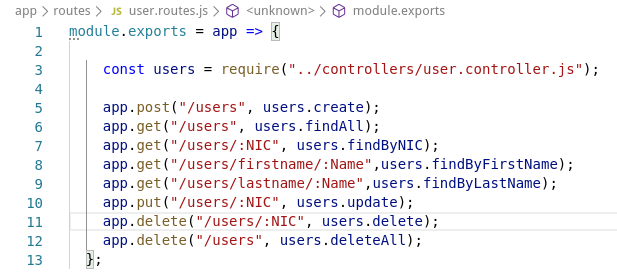


Figure XXI- user.routes.js

### Configuration

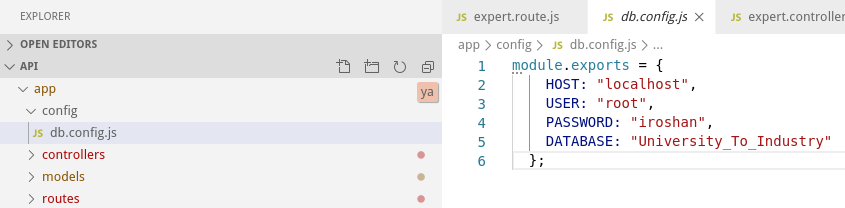


Figure XXII- Database Configuration

# Tools and Technologies

## Technology Used

To develop this project basically, we used the following technologies.

For the backend implementation

* NodeJS

Node.js is an open-source, cross-platform, javascript runtime environment that executes Javascript code outside of a web browser.[[1]](#footnote-1)

* ExpressJS

Express.js, or simply Express, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js[[2]](#footnote-2)

It is Back-end web application framework running on top of Node.js

* MySQL

MySQL is an open-source relational database management system (RDBMS). MySQL is written in C and C++. Its SQL parser is written in yacc, but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, and Tru64. A port of MySQL to OpenVMS also exists.[[3]](#footnote-3)

For the frontend implementation

* Vue Framework

Vue is an open-source Model–view–ViewModel JavaScript framework for building user interfaces and single-page applications.[[4]](#footnote-4)

Apart from above we used,

Bcrypt - which is a node.js password hashing method to hashing passwords.

Body-Parser - It simply let handle JSON objects in backend level.

## Why Did We Use Them?

* Node.js

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and effective, ideal for data-intensive real-time applications that run across distributed devices.[[5]](#footnote-5)

The following are some of the key features that make Node.js the first choice of software architects.

* **Asynchronous and Event-Driven**

All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the following API subsequent to calling it and a notification mechanism of Events of Node.js encourages the server to get a response from the past API call.[[6]](#footnote-6)

* **Fast**

being based on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.[[7]](#footnote-7)

* **Single-Threaded but Highly Scalable**

Node.js uses a single-threaded model with event looping. The event mechanism causes the server to respond in a non-blocking way and makes the server profoundly versatile as instead to conventional servers which make limited threads to handle requests. Node.js uses a single-threaded program and the similar program can offer support to a much larger amount of requests than conventional servers like Apache HTTP Server.[[8]](#footnote-8)

* **No Buffering**

Node.js applications never buffer any data. These applications simply output the data in lumps.[[9]](#footnote-9)

Express and Node Js both go combine manner in the development of backend. Those two interact with each other and shared their components to make interactions with frontends.

* Express.js

ExpressJS is a prebuilt NodeJS framework that can help you in making server-side web applications faster and more brilliant. Simplicity, minimalism, flexibility, scalability are some of its characteristics and since it is made in NodeJS itself, it inherited its performance as well.[[10]](#footnote-10)

In short, ExpressJS did for NodeJS what Bootstrap did for HTML/CSS and responsive web design.

* MySQL

While developing the database both MongoDB and MySQL were considered. But MySQL was used as it was asked specifically in the coursework documentation and due to the relational nature of the data that is used in the system, a relational database was the ideal database system and therefore MySQL was used.

* Vue

Vue is a progressive framework for building user interfaces. In contrast to other monolithic frameworks, Vue is designed from the beginning to be gradually adoptable. The core library is focused on the view layer only and is easy to pick up and incorporate with different libraries or existing projects.[[11]](#footnote-11) Then again, Vue is likewise flawlessly equipped for sophisticated Single-Page Applications when used in mix with modern tooling and supporting libraries.

Vue which permits you to put the HTML, CSS, and JavaScript inside to scope your component accurately. It is likewise valuable since you get fine aides like babel to deal with new syntax in JS like async/await.

## API Development Technologies

Mobile Application and Web Application use NodeJS REST API for controlling the database and applications. Rest API (Representational State Transfer) APIs are web standards-based architecture and use the HTTP Protocol for exchanging data between applications or systems. In RESTFUL web service HTTP methods like GET, POST, PUT and DELETE can be used to perform CRUD operations. REST is very simple compare to other methods like SOAP, CORBA, and WSDL, etc.

# Testing

## Web Application Testing

## Mobile Application Testing

# API Documentation

<https://documenter.getpostman.com/view/8511782/Szf81SyM>

A pdf of API documentation is attached with this project folder for more convenience.

# Instruction to Run

1. Navigate in to the API folder.
2. Open up a terminal.
3. Run 'npm init' to install all the dependencies.
4. Run either one of the following commands :

* -node server.js
* -nodemon server.js

# Contribution

## Workload Matrix

## LinkedIn Certificates

1. K C A A Iroshan 10638366

A screenshot of a cell phone

Description automatically generated

1. M D S Tharindu 10638387

A screenshot of a cell phone

Description automatically generated

1. T J M Siriwardhana 10638374



1. L T S Dassanayake 10638504



1. G H P Prabodhani 10638378



1. A A A Dulanja 10638431



# Turnitin Report

# References

Anon., n.d. *Algoworks.* [Online]   
Available at: https://www.algoworks.com/blog/why-use-expressjs-over-nodejs-for-server-side-coding/  
[Accessed 20 04 2020].

Anon., n.d. *tutorialspoint.* [Online]   
Available at: https://www.tutorialspoint.com/nodejs/nodejs\_introduction.htm  
[Accessed 20 04 2020].

Anon., n.d. *Vue.js.* [Online]   
Available at: https://vuejs.org/v2/guide/  
[Accessed 20 04 2020].

Anon., n.d. *wikipedia.* [Online]   
Available at: https://en.wikipedia.org/wiki/Node.js  
[Accessed 19 04 2020].

Anon., n.d. *wikipedia.* [Online]   
Available at: https://en.wikipedia.org/wiki/Express.js#cite\_note-4  
[Accessed 19 04 2020].

Anon., n.d. *wikipedia-MySQL.* [Online]   
Available at: https://en.wikipedia.org/wiki/MySQL#Overview  
[Accessed 18 04 2020].

Anon., n.d. *wikipedia-Vue.* [Online]   
Available at: https://en.wikipedia.org/wiki/Vue.js  
[Accessed 18 04 2020].

1. (Anon., n.d.) [↑](#footnote-ref-1)
2. (Anon., n.d.) [↑](#footnote-ref-2)
3. (Anon., n.d.) [↑](#footnote-ref-3)
4. (Anon., n.d.) [↑](#footnote-ref-4)
5. (Anon., n.d.) [↑](#footnote-ref-5)
6. (Anon., n.d.) [↑](#footnote-ref-6)
7. (Anon., n.d.) [↑](#footnote-ref-7)
8. (Anon., n.d.) [↑](#footnote-ref-8)
9. (Anon., n.d.) [↑](#footnote-ref-9)
10. (Anon., n.d.) [↑](#footnote-ref-10)
11. (Anon., n.d.) [↑](#footnote-ref-11)