**SCS 3214 / IS 3113: Group Project II - 2022**

**Group No: G32**

**Project Name: Entero (Event Management System)**

***Technology Justification***

***Front-End***

***Requirement of the selected technology***

* Should be free and open source
* Should support component-based development
* Supports faster development and light weighted rendering
* Should have a good community support and a good documentation

***Justification***

React JS and Angular JS both are open source. Angular JS has many built-in libraries for component but Angular JS is heavier because it is a complete framework and it is used for enterprise level products for small scale and non-enterprise level products it is much heavier.

React JS supports component-based development and it is better than Angular JS due to its virtual DOM implementation and light weighted rendering. We can hot-reloadthe system instead of recompiling while developing to see the results. In the limited timewe can develop the system quickly with React with the ability of reusable components.  Migrating between React JS versions is quite easy too you don't need to install updates one by one. The community that using React JS high so there is a good community support as well. With these reasons we choose React JS.

***Backend***

***Requirement of the selected technology***

* Having a good package manager
* Should be able to manage request and responses easily and efficiently
* Should require less resource with a good performance
* Should support in building RESTful API

***Justification***

Node JS has a good package manager (Node Package Manager) compared to Java and C# and also Node JS supports asynchronous processing, which is unique property that Node JS has, where the request and response is processed more efficiently. Both Node JS and JAVA supports in building RESTful API.

When comparing Node JS with JAVA and C#

|  |  |
| --- | --- |
| **Node JS** | **JAVA / C#** |
| Very fast buffering | Significantly slower |
| Highly scalable | No internal scalability |
| Asynchronous non-blocking IO. | Synchronous and blocking IO |
| Less resource is required | High Resource is required |
| A single thread can be used to handle multiple parallel request | Creates a thread per request |

According to the above comparison Node JS is better where it meets all the requirements needed.

***Database Management System***

***Requirement of the selected technology***

* Should be a Relational database management system
* Should be free and open source
* Should have a good documentation and community support
* Should support ACID properties
* Should have a good performance and flexibility
* Reliability when upscaling

***Justification***

In our proposed system there are clear relationships between the entity so using a relational database (SQL) will increase the performance than using a non-relational database (NoSQL). In that case as Mongo DB is a non-relational database management system as well as according to our requirement ACID properties should be supported but NoSQL databases (MongoDB) won’t strictly support ACID properties, accordingly it’s better to use relational database.

We have a chat component in our system we have planned to use relational database as the database. Most modern relational database systems can store, parse, and query "*unstructured*" data too, since that's really a subset of structured data. For example, the JSON capabilities of PostgreSQL that makes PostgreSQL a powerful data store for relational data and any additional meta-data that's unstructured as well.

Both MySQL and PostgreSQL are open source and free and both the database management system has a good documentation and a community support.

When comparing MySQL and PostgreSQL

|  |  |  |
| --- | --- | --- |
|  | **PostgreSQL** | **MySQL** |
| **DBMS type** | Object-based relational | Relational |
| **Upscaling reliability** | Very good | Fairly good |
| **ACID support** | Complete | Limited |
| **Join capabilities** | Good (Supports FULL (OUTER) JOIN) | Limited |

According the comparison in above table it concludes that the database management system which caters the system requirement is PostgreSQL.

***Backend Framework***

***Requirement of the selected technology***

* Should be free and open source
* Should have a good performance
* Should work well compatible with the selected database (PostgreSQL)
* Should support building RESTful API
* Should have a good documentation and community support

***Justification***

Express JS has a good documentation and it is a free and open source web application framework for Node JS. And Express doesn’t stick to a given structure when compared to Nest JS which uses MVC structure so flexibility in Express JS is high when compared to Nest JS where it will be easier when developing as Express JS has features like routing and middleware included in the framework. Through middleware, we can maintain access control for the user requests. With the limited timeline, it helps to finish the project on time. RESTful API built using Express JS works well compatible with PostgreSQL database. Express JS has number of plugins provided by the community which increases the performance and makes the framework more powerful and has a good community support. Accordingly, Express JS caters our technology requirement.