

|  |
| --- |
| Name: APSU Jayathilaka |
| Student Reference Number:  10749976 |

|  |  |  |
| --- | --- | --- |
| Module Code: PUSL3120 | Module Name: Full Stack Development | |
| Coursework Title: Full stack Development Project Proposal | | |
| Deadline Date: 16 Jan 2023 | | Member of staff responsible for coursework: Mr. Mark Dixon |
| Programme: BSc (Hons) Software Engineering | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed: | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |
|  | | |

# Project Proposal

## PUSL3120 Full-Stack Development

Proposal title: Online Ticketing site

Student number: 10749976

## Functionality

## This site provides a platform to buy tickets online for entertaining events hosted by Sri Lankan communities. Users can enter to the site, and they can surf events in nearby dates and locations. Events will be categorized into multiple categories according to the type of it. Like Musicals, Exhibitions, Concerts and Drams etc. Event organizers can connect admin through the web site and request a quotation price for adding events. After quotation organizers can publish their events in the site. Events contain difference classes, ticket prizes, requirements as well as offerings like features So organizers can publish those features according to their events. Users just have to select the event and fill their requirements in the events. When complete the transactions user will receive a mail or message early requested credentials. Then user can use that in event day to enjoy the show. Admin can create, read, delete and update events. They can monitor stats of selling tickets.

## Required properties

### Interactive

Normal Users

Normal users can enter to the site and select their event to buy tickets. After selecting event, page will redirect to booking page and user need to fill some options related to ticket categorize. And that page contains all the details about the event. After proceeding the booking page, user need to add some of his credential to send the digital ticket copy. After that user can pay the bill in payment gateway and complete the booking.

Event Organizers

Event organizers can request a quotation price through the Site and from that they can interact with admins to publish an event

Site Admins

Site admins need to login to the site entering username and password. And they can create, read, update and delete events according to organizer’s requirements. As well as can monitor the ticket sales.

### Database

For the database facilities this site use Mongo DB. In client interactions all the events and related data will store in databases. As well as Database use to update the ticket counts. If not, it can crash the seating facilities. In admin side database table monitorization facility will be available with CRUD operation.

### Web Sockets

### Client and admin can make conversations like a customer care Service. To that web site provide a chatting facility in user side. So, to do that it need to use web sockets. As well as hope to use for passing the booking data to place the reservations.

### Structured

## This web site is basically designed to provide services to buying tickets related to different kind of entertainment events. So here CRUD operation will take place. User authentication will be used. MEAN Stack technology will use with MVC architecture for backend frontend and databases. So, classes will allocate according to basic operations. Models go with OOP concept models files.

## Planned Work

### Resources

All the Development phases are doing with MEAN Stack technology. Frontend development do with angular framework. Server-side coding, web APIs and web sockets will be doing along with node.js on Express server. MongoDB is the database use to store all events information and sales.

### Testing

This site will be tested with the unit testing, integration testing, system testing and accepting testing. According to the developments and modules it will lead the tests.

Unit Testing – Executes unit codes and check whether the expected result is received. Form Validations, responsiveness of the pages and links will be tested. All backend modules API implementations and front end and backend connections will be tested.

Integration Testing – in this phase functionality and accuracy within two or more units will be tested. Later the frontend backend unit tests done, site will again test to backend APIs and front-end connection.

System Testing - End to end system will be tested for system specifications. After the unit tests and fixing errors, again it tests to check all CRUD operations.

Acceptance Testing- In here site will be tested to secure quality and user experience. As well as check all the business requirements to ensure the process.

### Work Plan

Timeline

Description automatically generated