1. EXECUTIVE SUMMARY

In this project, we propose to build a web application for people who are struggling to find the details when planning their own events and offer our own niche services. It will provide popular sought after options in regarding essential event types. The web application is working as a one-stop shop which provide multiple services for their clients in one place. The purpose of this one-stop shop is to provide a convenient, efficient and cost-effective service for their customers. We will be incorporating basic HTML, CSS, MySQL, Node.JS and Javascript to develop this web application. This is a self-sponsored, group project utilising the agile method, where we have created user cases to provide a solid platform to construct our event website.

|  | **Project Parts** |  |  |
| --- | --- | --- | --- |
| **Brendan Cen** | Services Page (HTML, CSS, Javascript) | Booking Page (HTML,CSS) | Database  Database pages (HTML, CSS, Javascript) |
| **Sachini Waduge** | Home Page (HTML, CSS, Javascript) | Gallery Page (HTML, CSS, Javascript) | Feedback Page (HTML, CSS)  Admin Page (HTML, CSS)  Database Bookings Edit/Delete Page (HTML, CSS, Javascript)  Database |
| **Puneet Puneet** | Login Page (HTML, CSS) | Register Page (HTML, CSS) | Contact Us (HTML, CSS)  Database  Repository Flow  Project Management  GitHub |

1. INTRODUCTION

In modern world, people are busier with their day-to-day lifecycles, therefore if they have a special occasion, such as birthdays, get-togethers, gender reveal parties or any other occasions, people do not have enough time to allocate for those additional tasks due to busy lifestyles. The main reason being, so far in the market there is no such opportunity that people could go to one single platform and find all the relevant services as per their needs. (John Allen, 2000)

This web app will also feature a database of information of all the parts of the chosen services. Target users will be the parents, working professionals, young couples and people who need to organize their own events. Anyone can search for the details on the website but if they want to make a booking through the website, the user will have to register on the app before proceeding with the booking phase.

The expected outcome of this web application is to showcase our own services to the users in one place and provide a one-stop shop solution for the end users. Furthermore it will help to reduce tension of people and improve the customer experience because the users will be able to quickly search for our specific service selection relating to the events we have available.

Therefore, we believe that this web application will be helpful for people who need to organize their events more easily and from the other hand give us a platform to promote our own niche services to the users and build our company brand.

INITIAL PROPOSAL

1. INTRODUCTION

Events and festivals have become strong demand generators and plays a significant role in todays world with busy lifestyles of people. In modern world, people like to celebrate their special moments with their loved ones, families and friends. Due to their busy lifestyles even if they want to celebrate their special moments, they are facing problems of making time to plan their moments. For example, to celebrate their birthdays, newborn children, gender reveals, kids’ birthday parties, weddings, anniversaries, get togethers, conferences and other celebrations. Reason being, so far in the market there is no such opportunity that people could go to one single platform and find all the relevant services as per their needs.  Hence if they have a special event, either they have to go to different places, ask from different people or they have to search details in different websites to gather information or they have to search and hire an event planner to organize the event by paying an extra cost even if it is a small event (John Allen, 2000)​. ​

In this project, we propose to build a web application for people who are struggling to find the details when planning their own events. It will provide all the information in one place regarding essential parts of the event. The web application is working as a one-stop shop which provide multiple services for their clients in one place. For example, venues, catering services, decorations, music and other services. The purpose of this one-stop shop is to provide a convenient, efficient and cost-effective service for their customers.  Further it will help to reduce tension of people and improve the customer experience because the users will be able to quickly search for all information relating to every step of the event around them.

This web app will also feature a database of information of all the parts of the event. Target users will be the parents, social clubs, working professionals, young couples who are planning their weddings, event planners and the people who need to organize their own events. Anyone can search for the details on the website but if they want to make a booking through the website, the user will have to register on the app before proceeding with the steps. Further the service providers also can register on the website by creating a login and then the end users can see the details of those service providers.

The expected outcome of this web application is to gather all service providers and users into one place and provide a one-stop shop solution for the end users. Therefore, we believe that this web application will be helpful for the people who needs to organize their events easily and from the other hand the service providers can use the web applications as a marketing tool to sell their products and services to their end users.

2. SCOPE

The final project will have four main informational webpages including a home page, services page, register/login page and finally the contact page. The home page will provide all the information about the web application and categories of events. Further it will navigate to the next pages as well. The second page will show the details about the offering service of the registered service providers. End users can select the services according to the nature of their events. This is the page for most of the end users to find their details and the page for the service providers to market their products and services. Therefore, any user can access the information which is available in the web application, and they can easily filter and search the required details by selecting the preferred party type and the suburb.

The next web page is for registration/login for both end users and to the service providers by providing the details. If the end user wishes to proceed with the service, they can login to the web page and make the booking through the website. If the service provider likes to sell their products and services to the customers, they can login to the website and upload their details and photos to the website to promote their services and products. Finally, there will be a contact page for all users to submit their comments, feedbacks and queries to the website.

The final project will also have 3 main databases. One is for all the information of registered service providers, which will be called upon when someone is searching for details. Second database for registered users/login info that will be called up when someone is adding new register details or logging in. Final database will hold all the contact information, comments, feedbacks and queries which submitted by the users to the website.

3. USER STORIES AND CASES

As a user I need to search the details to organize a party event in a particular location in Auckland so I can plan my event more easily by selecting the important components of the event. The user can select the party type and then filter the services in service page by selecting the location. Finally, we will test the search function randomly by selecting different locations and services.

The service agent also can register on the web site and add their details to the website. Users can register on the website by providing the basic information. This page also will test by registering users.

As a user I want to be able to contact the site so I can ask questions or provide feedback. This process should send off an acknowledgement to the user by saying that their query has been sent & we will get back to you as soon as possible. This page also will be test by submitting different feedbacks.

3. CURRENT USER STORIES AND USE CASES

As a user I need to be able to book an event through selecting the services for organising an event so I can plan my event more easily. The user can choose each desired service type and then filter the services choices in accordance to their event requirements.

As a user I need to be able to register on the website in providing my basic information to set up a profile, so I can have access to the website functions including booking and giving feedback. This page also will be tested by registering new users.

As a user I want to be able to contact the website so I can ask questions/queries regarding the events and services. This will be carried out by testing the questions form for random users then showing the acknowledgement message that your questions/queries have been reached and we will get back to you as soon as possible.

As a user I want to be able to provide feedback after having my event, where I can give my input on what improvements need to be made to the company in providing better service. This process should send off an acknowledgement message to the user by saying that their feedback has been sent & we will get back to you as soon as possible. This page also will be tested by submitting different feedbacks.

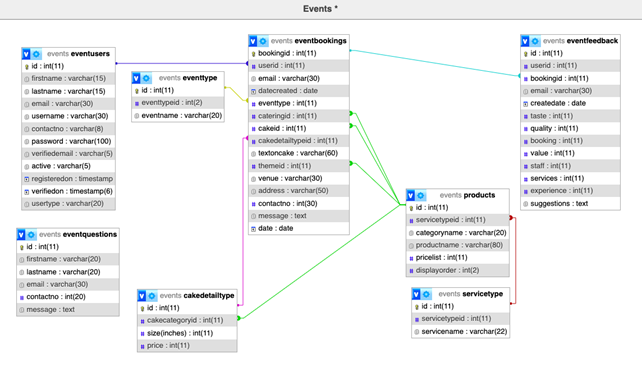
As an admin I want to able to view database tables for editing or deleting data so I can manage the bookings in a convenient manner. This will be carried out by testing various registered users and modifying changes to the booking page to ensure they can load correctly and the edit/delete functions are working properly.

4. VARIATIONS IN USER REQUIREMENTS FOR INITIAL PROPOSAL

For our project, we have made some major adjustments in fitting with the new idea to represent a niche event services start up company that include:

* Removing the event venue and location function in favour of promoting our own niche events and services to the user. The user would have to already have booked a venue and then venture to our website to add on services in accommodating their event plans.
* Remove the service provider page as the group felt this was an overload of functionality and being a start up company, it would be better to focus on selling our service. We wanted to put the main attention on the features of our services for the user coming to the website.
* No service provider register or login page. We felt it was unnecessary at this stage of our company to include the service provider, simply wanting to divert more time and effort to showcase our services, where we are the main service provider.
* Limiting the number of event types to 4 events, therefore popular events such as weddings and conferences would be excluded. The user will see that there are 4 events available to choose from in the website.
* Limiting the number of service types to 3 services. Staying true to our niche start up company, we will be only focussing on 3 main services which are catering food, cakes and birthday theme set up. The user will not be able to locate all essential wedding services but more pinpoint to small scale events for our service selection.
* Adding a gallery page to show the flavour of our company through the past events we have been done, in a colourful fashion and give the users a good indication of how our services are implemented to create a vibrant experience for the users event.
* Increase in database tables to 7 in total, due to the number of added functions, nature of our services and overall event structure of our application. We need to incorporate more tables to show effective relationship between users and the services they use.
* Adding the admin functionality, to enable the admin to monitor all bookings information from the registered users in case of any customer changes in booking requirements and more importantly view the database tables to show all feedback, questions/queries and registered users details.

5. DESIGN DOCUMENTS



DATABASE SCHEMA

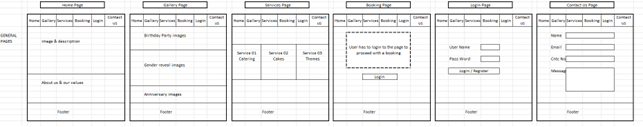
Our database schema has a total of 8 database tables, which are event users, event bookings, event type, service type, products, cake detail type, event questions and feedback in making the foundation of our database structure. The event users table is the main section of the database for registering and login to clearly identify the new users who are registering in the website and at the same time the users wanting to login so they can make a booking.

In the booking database, there is a expansive number of fields relating to the booking sections of the form covering event type, catering id, cake id, theme id, cake detail type id, text on cake, venue, address, contact number, message and event date. These are all the necessary information we need to gather from the registered user to complete the event booking. The event type is linked back to the event type table, whereas the identification for catering, cake and theme are foreign keys that branch back to the primary key of ID in the products table to distinguish the chosen service products effectively (IBM, 2021).

For the event type, it labels the specific event categories in a simplified format, and the service type follows the same structure. This allows the user to know what event and service type they have chosen from the bookings table. There is a products table showing the service type id, 3 service categories, all product names, pricing of the 2 services excluding cake that is found in a separate table. For the cake detail type, there is the cake category id referring to the cake choices from the products table, associated to their cake sizes (6 inches, 8 inches and 10 inches) and respective prices.

The questions database is located in the “Contact Us” page. Compare to the other tables, this is unique as the questions/queries database is applicable for all registered and non-registered users, therefore any user who comes across our website can ask us questions. The questions table stands alone as anyone can access the questions form without linking back to the users table. Lastly is the feedback database, strictly available for only registered users who have already finished having their event and the feedback form is encouraged to be filled out to help check our service performance in providing recommendations in improving the overall user experience. This is linking back to the bookings database to show the direct correlation of the user booking the event then having the opportunity to give us feedback in following up from the event proceedings.

Software Prototypes



A white sheet with black lines and letters

Description automatically generated with medium confidence

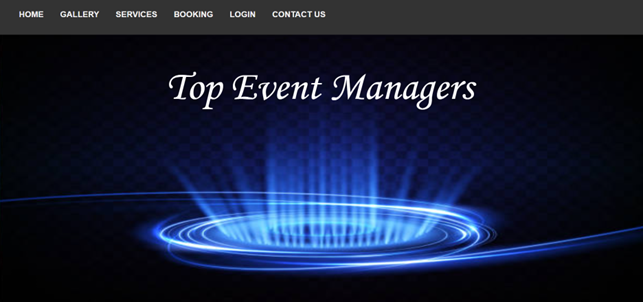
A white grid with black lines and letters

Description automatically generated with medium confidence

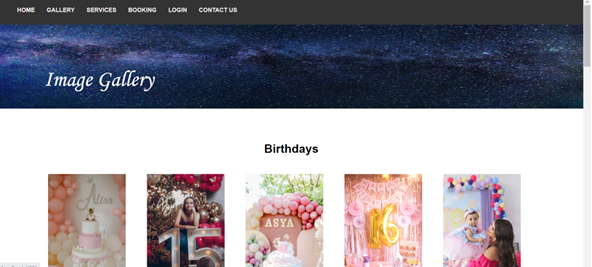
**Software Pages**

**General Pages**

General Home Page



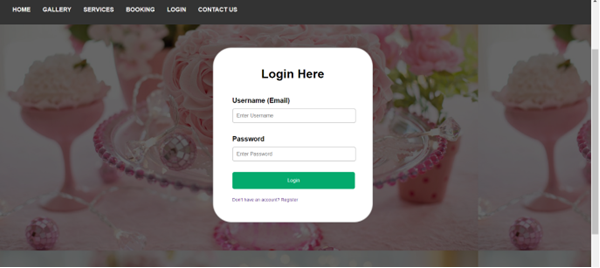
General Gallery Page



A screenshot of a computer

Description automatically generated

General Services Page



General Login Page

**User Pages.**

User Booking

A screenshot of a computer

Description automatically generated

User Feedback

A screenshot of a computer

Description automatically generated

User Booking Summery

**Admin Pages**

Admin Booking Summery

A screenshot of a computer

Description automatically generated

Admin Users Summery

A screenshot of a computer

Description automatically generated

Admin Queries

A screenshot of a computer

Description automatically generated

Admin Feedback

A screenshot of a computer

Description automatically generated

(Puneet add Gantt Chart)

6. SOFTWARE DESIGN

For our website we have implemented many pages of source code to fully demonstrate the coding structure of the event management website.

The main element of our app code is (server.js), this is the core engine of the code from the server’s point of view. It orchestrates and conducts the operation of every page of our web application, and is directly involved in setting up the database functions. We used Javascript language to code the key server file. With the help of Javascript source codes, we can bring basic functionality of our website application to life in making our pages more user friendly (Future Skills Academy, 2023).

The first standard code we have established is found on the server.js file. There is a strong presence of this type of code, which calls on the home page to be opened. For each of our pages “Home, Gallery, Services, Booking, Login, and Contact Us”. This coding line is employed from the server.js to be able to display on the website.

app.get('/', function(req, res, next) {

res.render('home', { title: 'Home' });

});

The files for userregister.ejs, login.ejs, Booking.ejs, contactus.ejs and userfeedback.ejs have direct contact to insert or retrieve data into mySQL database. The different page source codes are shown below:

Registration Form Submission

In userregister.ejs, using the “form action=“/register” in simple terms emits the required action that is called when the form submit button has been activated (Future Skills Academy, 2023):

<div class="container8">

<form action="/register" method="POST">

<h1>Register Here</h1>

<label for="firstname"><br><b>First Name</b></label>

<input type="text" placeholder="Enter First name" name="firstname" required>

<label for="lastname"><br><b>Last Name</b></label>

<input type="text" placeholder="Enter Last Name" name="lastname" required>

<label for="email"><br><b>Email</b></label>

<input type="text" placeholder="Enter Email" name="email" required>

<label for="username"><br><b>Username</b></label>

<input type="text" placeholder="Enter Username" name="username" required>

<label for="phonenumber"><br><b>Phone Number</b></label>

<input type="text" placeholder="Enter Phone Number" name="phonenumber" required>

<label for="password"><br><b>Password</b></label>

<input type="text" placeholder="Enter Password" name="password" required>

<br>

<button type="submit">Register</button>

</form>

</div>

By clicking the “submit” button, the server is immediately notified the required action has been carried out then sends the code line app.post(‘/register’..) to function. The purpose of the function response is to take the information the user has given in the form and place it into the event users table stored in mySQL database (Future Skills Academy, 2023).

app.post('/register', function(req, res, next) {

var firstname = req.body.firstname;

var lastname = req.body.lastname;

var email = req.body.email;

var username=req.body.username;

var contactno=req.body.contactno;

var password = req.body.password;

var sql = `INSERT INTO eventusers(firstname, lastname, email, username, contactno, password) VALUES ("${firstname}", "${lastname}", "${email}", "${username}", "${contactno}", "${password}")`;

db.query(sql, function(err, result) {

if (err) {

throw err;

res.render('register', {success: false, errormessage: 'Something went wrong!'})

}

else {

console.log('1 record inserted');

res.render('login');

}

});

});

It is essential to understand the importance of the following SQL statement from the app.post(‘/register..) code. The statement acts as a linking bridge to send the users registration details to the database, so it can be retrieved for the admin to verify the users’s credentials when they are creating an account (Future Skills Academy, 2023).

Showing the Users Registration Data

To be able to view all users’s registration information, the code app.get(‘/geteventusers’….) is called upon, and its role is tabulating all the user’s data from the events users table stored in the database, puts the retrieved data into “eventusersData” and project the data onto the localhost3000:/geteventusers page (Future Skills Academy, 2023.

app.get('/geteventusers', function (req, res) {

db.query("SELECT \* FROM eventusers", function (err, result) {

if (err) throw err;

console.log(result);

res.render('geteventusers', { title: 'List of Registered Users', eventusersData: result });

});

});

Since we have been taught to work with “ejs” file, the eventusersData uses the ejs structure by applying Javascript code. The code is operating between the two percentage signs symbols <% %> (Future Skills Academy, 2023).

We are able to retrieve eventusersData to be drawn up as a HTML Table because we have employed the if/else loop statement shown below (Future Skills Academy, 2023):

<% if (eventusersData.length !=0) {var i=1; eventusersData.forEach( function(data) {

%>

<tr>

<td class="login2"><%=data.id %></td>

<td class="login2"><%=data.firstname %></td>

<td class="login2"><%=data.lastname %></td>

<td class="login2"><%=data.email %></td>

<td class="login2"><%=data.username %></td>

<td class="login2"><%=data.contactno%></td>

<td class="login2"><%=data.password %></td>

<td class="login2"><%=data.verifiedemail %></td>

<td class="login2"><%=data.active %></td>

<td class="login2"><%=data.registeredon %></td>

<td class="login2"><%=data.verifiedon %></td>

<td class="login2"><%=data.usertype %></td>

<td class="bookingdata3">

<td class="bookingdata3">

<button class = "button button2"onclick="deleteRow('<%= data.id %>')">Delete</button>

<form id="deleteForm" action="/geteventusers" method="post" style="display: none;">

<input type="hidden" name="id" id="IdInput">

</form>

</td>

</tr>

<% i++; }) %> <%}%>

Submit Login Data

When the user has completed the registration form, the next phase is to complete the login form and submit the login information. Using login.ejs, the code for “form action=“‘/login” will signify after the user has activated the submit button, the action will be called to start the app.post(‘/login’..) function (Future Skills Academy, 2023).

app.post('/login', function(req,res){

var user\_name=req.body.user\_name;

var passw=req.body.passw;

db.query(`SELECT \* from eventusers where email="${user\_name}"and password="${passw}" `, function(err,result){

if (err) throw err;

if (result.length > 0) {

var usertype = result[0].usertype;

if (usertype === 'Admin') {

res.render('adminhome');

} else {

req.session.usertype = 'User';

res.render('userhome');

}

} else {

res.render('loginerror');

}

});

});

There is a noticeable difference to the code structure for login compare to the register section, where rather than inserting data into the database, admin is checking all the event users in the event users table to verify if the user matches the email and password they have provided. Therefore when the user clicks on the login button, the app.post(‘/login’..) code is initiated.

It takes the SQL statement of “Select” all the users from event users table through checking the specific user information given from the login form in order to validate whether the user has the correct existing email and password to confirm the verification process. If the credentials match with the database then the user has successfully signed into the website and will be directed to the user home page (Future Skills Academy, 2023).

Display Login Users Data

This is the same step as initiating the app.get(‘/geteventusers’..) to bring up the database for the event users data, which will also include the login information for all the users with their username and password being visibly displayed. When the admin clicks on the users link at the top menu of the home page, it will direct them to show the event users data in a HTML table that has all the login details available for the admin to fully view and check validation (Future Skills Academy, 2023).

Display Bookings Data for Login Users

app.get('/userlogin', function (req, res) {

db.query(`SELECT eb.bookingid, concat(eu.firstname, ' ', eu.lastname) as username, eb.email, eb.datecreated, eb.eventtype, et.eventname, p.productname as cateringname, eb.cakeid, eb.cakedetailtypeid, pp.productname, ct.size, ct.price, eb.textoncake, eb.themeid, ppp.productname as themename, eb.venue, eb.address, eb.contactno, eb.message, eb.date FROM eventbookings eb

left join eventusers eu on eu.id = eb.userid

left join eventtype et on et.eventtypeid = eb.eventtype

left join products p on p.id = eb.cateringid

left join cakedetailtype ct on ct.id = eb.cakedetailtypeid

left join products pp on pp.id = ct.cakecategoryid

left join products ppp on ppp.id = eb.themeid`, function (err, result) {

if (err) throw err;

console.log(result);

res.render('userlogin', { title: 'List of Event Bookings', userbookingsData: result });

});

});

The bookings information for the registered users is called upon after the app.get(‘/userlogin’..) pulls the data of all event bookings in the database then presents the retrieved booking data in the form of a HTML table. This is brought up when the logged in user opens the login link in the top menu to display the booking summary of all the event bookings (Future Skills Academy, 2023). However, the key difference is the booking table fields for specific services are linked to multiple tables, and we need to extract the product name values by implementing a recommended “Left Join Statement” from peer tutor discussion (W3schools, n.d.). In order to show the correct details related to the services in the event bookings table, we are joining multiple tables to retrieve the necessary data from event type, products and cake detail type so we can get the accurate booking information to display to the admin for them to see the final results (W3schools, n.d.). The short abbreviation represent the table names for event users, event bookings, cake type for cake detail type and products. This coding structure is extremely complex but its main aim is to source the required information from multiple tables for the booking data to be precise and show the product names for the admin to understand in a simplified form.

Display Bookings Data for Login Users

app.get('/getbookings', function (req, res) {

db.query(`SELECT eb.bookingid, concat(eu.firstname, ' ', eu.lastname) as username, eb.email, eb.datecreated, eb.eventtype, et.eventname, p.productname as cateringname, eb.cakeid, eb.cakedetailtypeid, pp.productname, ct.size, ct.price, eb.textoncake, eb.themeid, ppp.productname as themename, eb.venue, eb.address, eb.contactno, eb.message, eb.date FROM eventbookings eb

left join eventusers eu on eu.id = eb.userid

left join eventtype et on et.eventtypeid = eb.eventtype

left join products p on p.id = eb.cateringid

left join cakedetailtype ct on ct.id = eb.cakedetailtypeid

left join products pp on pp.id = ct.cakecategoryid

left join products ppp on ppp.id = eb.themeid`, function (err, result) {

if (err) throw err;

console.log(result);

res.render('getbookings', { title: 'List of Event Bookings', bookingsData: result });

});

});

This is the same foundation code for showing all the bookings data, that is retrieve from the app.get(‘/geteventbookings), which assembles the users booking details in the database then viewed by the admin in a HTML table. The admin can access this booking report page after they login and click on the bookings tab in the top menu (Future Skills Academy, 2023). Employing the same technique with the “Left Join Statement”, the event bookings tables is joining the required tables in linking the necessary information, extracting it from those tables data, to bring up the product name values specific to each service, in projecting the exact information for the multiple services and selected tables in the booking database (W3schools, n.d.).

Display Questions and Feedback Data

For the questions, it calls upon the app.get(‘/getqueries) to bring up all the user questions from the database library and the admin can look through the questions HTML table by opening the queries link in the home page. This is the same formula for feedback, in activating the app.get(‘/getfeedback) function, the data for the feedback table will show the users feedback in HTML form after the admin accesses the feedback link in the homepage to get the feedback details (Future Skills Academy, 2023).

Express Session for Viewing Database Pages for Admin

In this section, only the admin has access to fully view the booking information of all registered users through implementation of the express session program. The first command after installing the express session is to add the code from line 8-12 in server.js with the required elements to start the session and create a secret key only for the admin to remember (NPM, 2021).

app.use(session({

secret: 'this is my secret',

resave: false,

saveUninitialized: true,

}))

Secondly, in the app.post(/‘login..), the first line of session code is demonstrated for line 284, which is indicating set up the session for the user type that looks at the SQL statement to obtain the records from the event users table in the database to see if the details are stored (NPM, 2021). If the details are clearly matching the user login then they will be granted to the home page (NPM, 2021).

app.post('/login', function(req,res){

var user\_name=req.body.user\_name;

var passw=req.body.passw;

db.query(`SELECT \* from eventusers where email="${user\_name}"and password="${passw}" `, function(err,result){

if (err) throw err;

if (result.length > 0) {

var usertype = result[0].usertype;

if (usertype === 'Admin') {

res.render('adminhome');

} else {

req.session.usertype = 'User';

res.render('userhome');

}

} else {

res.render('loginerror');

}

});

});

The main objective of the session code is to give 2 conditions. The code is checking whether the user type is actually logged into the website, where if they are unsuccessful then the result is undefined for the user type status (NPM, 2021). The second condition, checks to see if the user that has logged in is the correct admin type from the event users table, therefore if the admin details are the same as the login information the session will be created (NPM, 2021).

Using the bookings data as an example, the bookings page is now accessible for the admin to view all booking details and make any required changes as per the customers request. If the login credentials differ from the admin details in the database, it will immediately redirect the user to the login page, restricting access to the database pages (NPM, 2021). Only the admin will have inside knowledge to access the data tables with feedback, questions/queries and registered users personal information in keeping track of records, cancelling unwanted bookings and responding to any customer preference alterations.

app.get('/getbookings', function (req, res) {

let sess = req.session;

if(typeof sess.usertype != 'undefined') // is the user is logged in or not

{

if(sess.usertype === 'Admin') // is the user is logged in and the user type = Admin

{

db.query("SELECT \* FROM eventbookings", function (err, result) {

if (err) throw err;

console.log(result);

res.render('getbookings', { title: 'List of Event Bookings', bookingsData: result });

});

}

else

{

res.redirect('/login');

}

}

else

{

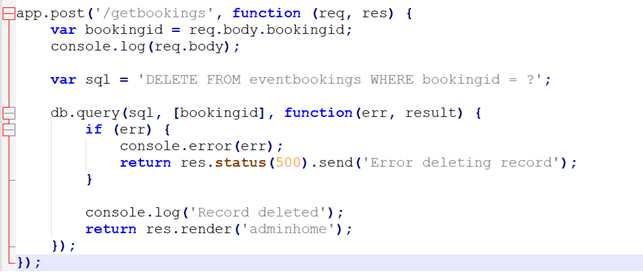
res.redirect('/login');

}

Delete function for Admin Database Pages

The purpose of the delete function is to facilitate the removal of records from the eventbookings database table based on a specified booking ID. If user requested to cancel a booking, the admin can login to the web application and delete the relevant booking from the eventbookings table. The delete function enhances the functionality of the event management system by providing a mechanism to remove bookings, thereby ensuring the accuracy and relevance of the stored data. Careful error handling is implemented to maintain the robustness of the system, and successful deletions are logged for monitoring and audit purposes.

The delete function is implemented as an HTTP POST request to the '/getbookings' endpoint. It retrieves the booking ID from the request body and constructs a SQL DELETE statement to remove the corresponding record from the 'eventbookings' table.



7. DEVELOPER DOCUMENTATION

TOOLS AND TECHNOLOGIES USED (IN ALPHABETICAL ORDER)

* CSS
* EJS
* Express.js
* HTML
* Javascript
* MySQL
* Node.js
* Notepad++
* phpMyAdmin
* Visual Studio Code
* XAMPP

8. OVERVIEW OF SOFTWARE AND ITS SOURCE CODE

For our final source code, you can locate the “EventManagement666” file under the nodeapps(?) file directory, where the root begins in C. Drive.

We have given the app source code the name “EventManagement666”. Inside the folder, you will find the following files, they are dbConfig.js, node\_modules folder, package-lock.json, package.json, public folder, server.js and views folder. There is a code file called “dbConfig.js” that basically configures the main database to store all the event data under the database name “events” (IBM, 2023).

For this project we have not touched the next 3 files, but the node\_modules folder illustrates the directory of npm setup to store all installed packages (StackOverflow, 2023). The package-lock.json purpose is storing installed or updated packages that are automatically written by npm and lock their versions in place while package.json purpose is managing and documenting all project information including name, version, author, dependencies and other relevant configuration data (Medium, n.d.).

Clicking on the public folder will show two types of folder, the first is images folder that contain all the used photos implemented in the final website layout. The second folder refers to the stylesheets folder, inside there are two types of files, one is style.css. This file is the central styling system to modify the main pages of our website excluding the booking page. The second file, style1.css is set up specifically to create the booking page style, which is different to the other page design layout hence the reason for making two stylesheets.

Our main file is the server.js, which assimilates the construction and development of each page of the website and is essential to set up the database pages.

The last folder called views, stores all our important files for designing and coding the event website from start to completion. The admingallery.ejs highlights the gallery page for the admin to see, adminhome.ejs is the home page for the admin, and adminservices.ejs shows the services page to the admin. Next there are two booking pages, one consists of the booking page “Booking.ejs”, activated when the user has logged into their account and the other “booknow.ejs” focusses on the booking page when the user has not logged in.

For the cakes.ejs, it shows the services page for the cakes section, catering.ejs is the catering page for our menu selection, and contactus.ejs is the contact us page for any user to ask questions or queries, There is the getbookings.ejs designed for the admin to see the event bookings table, geteventusers.ejs is the admin page to view and verify registered users details, getfeeback.ejs is the admin page to see all user’s feedback, and getqueries is the page for the admin to see all the questions and queries. The home.ejs is the home page when the user enters our website, login.ejs is the page to show the login form, and loginerror.ejs shows the login form when incorrect details are filled out and it will display this message.

The register.ejs shows the register form for the user to register, services.ejs is the main service page to show the 3 services, theme.ejs is the page when they click the image photo for the theme it will direct them to this page to show all our theme options, usercakes.ejs directs them to the cake page, usercatering.ejs shows the catering page to the registered user, userfeedback.ejs is the feedback page for the registered user to give us feedback, usergallery.ejs shows the gallery page for the registered user, userhome.ejs shows the homepage with the “User Login” to show they have successfully logged in, userlogin.ejs shows the page with all the confirmed event bookings information for all users, and user services.ejs shows the service home page for the registered user.

Under the views folder, the EJS format files will show the webpages to be displayed in our event website.

To commence the app process, open the XAMPP software, which controls the servers for Apache and MySQL (EDUCBA, 2023). Start the servers, after successfully showing both servers buttons have turned green, enter in the command prompt node followed by server. This should show:

**C:\nodeapps\EventManagement666\node server**

It is essential to carefully examine the source of the app file location before proceeding to enter the information into the command prompt.

If the app is running smoothly with no error, you should be able to see the following message on the command prompt:

**C:\nodeapps\EventManagement666\node server**

**Running at Port 3000**

**Database is connect successfully!**

The final phase is to open a new browser, input localhost:3000 into the address bar and the main homepage will appear on the users screen. You are now freely able to navigate different areas of the website, however there are some restricted sections of the website for booking and feedback that require the user to be registered and logged in to their account before they have full access.

In order for the admin to locate the user’s booking information, open the top link showing bookings, where the admin can now view the event bookings list to edit or delete changes.

9. KNOWN BUGS AND LIMITATIONS

The most common known bugs is from the back-end side of the website development for our coding, we were constantly getting error messages related to “cannot get” and “cannot post” error. As we were moving back and forth with the water safety app files and our own files, sometimes the updated code would be stored in the wrong file folder, which meant many times the browser couldn’t pick up the information to be displayed on the website page. Working with too many folders for one project is a huge bug that causes disorganisation and prone to error messages appearing frequently if not working with the right folder.

Another issue is when the user has registered, logged in and completed the booking form then all the booking information of every registered user is available for browsing. Not only can they see the other person’s personal information but they could exploit it for negative reasons. This is also a breach of privacy as giving personal information such as phone number and email should be hidden from others, and shows a weak security system that could lead to hackers stealing information and taint our company image in terms of safety and assurance of protecting users information.

If the registered user has any changes to their booking plans, the only way they can modify is through the action of the admin by sending a request in the questions/queries or through admin’s email address. There is no current option for the user to manually change or cancel their booking plans, this presents inconvenience for the registered user having to reach out to the admin by filling in the contact us form in order to make adjustments to their booking.

There is no payment page for the registered user to finalise the payment procedure. This makes the booking process feel incomplete and the only solution is to wait for an email from the catering company to guide them through the payment instructions. This is not within the website functionality but using the email source to finish the payment process, therefore for the user it is highly inconvenient to go through with the booking payment in delaying the successful booking confirmation.

For our catering services, we have fixed menu packages set for the users but if they want to modify their menu selection, delete some options or suggest their own dishes in the menu construction, we do not provide this functionality presently. We have created unique menu packages and can only allow the user to choose the contents. The limitation of menu personalisation can frustrate a user who may have different views of our menu selection and for guests who have special requirements pose a massive problem to not cater to their delicate needs. Also there are limitation in excluding drinks, venue and location from our service options, as the user can only choose 3 services where more popular services are not provided by us and can make the user feel like they are not getting the full event booking experience.

A visible limitation in our catering section, is we are incorporating different menu options using the name “Menu 1” as the base name. This is setting the options in the default way, which can be difficult for the customers to identify clearly the final menu package they will be selecting when filling out the booking. The plain standard looking “Menu 1-9” makes it feel outdated and some customers might have a hard time remembering the menu name, where they would have to transfer back to the catering page to recall their catering choice, prolonging the booking process.

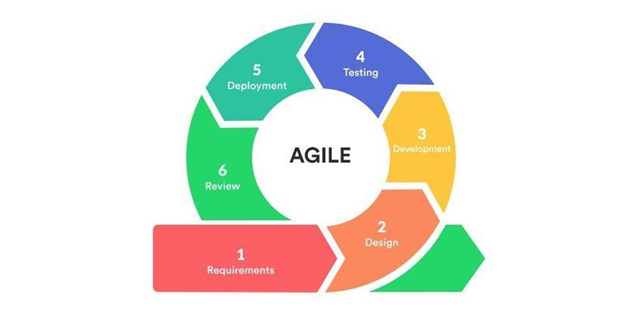
In the event database, there is a major error with testing the booking form submission where the service values chosen don’t match up accurately with the updated booking. From the admin point of view, the cake id, cake detail type id and theme id all point to the menu options rather than the specific service product. This makes it extremely inconvenient for the admin to figure out the exact booking information, as the values are not clearly classified. The cake detail type id mimics the size of the cake field in the booking form, but it doesn’t reflect the correct values when checking the cake detail type id table, making it an unfavourable task for admin to lock in the final booking confirmation.

These daunting errors, show the clear bugs when passing the form information onto the admin for clarification of all service products, which can lead to slow updates for the registered users to get assurance of their booking confirmation, and constant rechecking to make sure the service products are identical to the booking. In addition, the admin needs to look at the message section of the booking table to locate the number of guests, therefore they take more time to check important details in an unorganised format rather than showing the main points of the booking process.

10. SDLC

Wechoose Agile Methodology to develop our web application because the Agile model helps teams identify and address small issues on projects before they evolve into more significant problems, and it engages business stakeholders to give feedback throughout the development process (Target Trend, 2021). The journey of the progression of our project had countless of hiccups but it was extremely beneficial incorporating the agile method to be able to take an aspirational project break it down into small sprint sections to make necessary changes when there were obstacles hindering our development and get continuous feedback to keep us moving forward (Target Trend, 2021). We feel this model acted like a resourceful road map to always guide us when we were having a tough time to assemble the functionality piece by piece and we would have great assurance to implement this methodology in future projects.

\*Retrieved from (Sachini please reference this diagram and check is it the right reference as I see multiple agile method references)

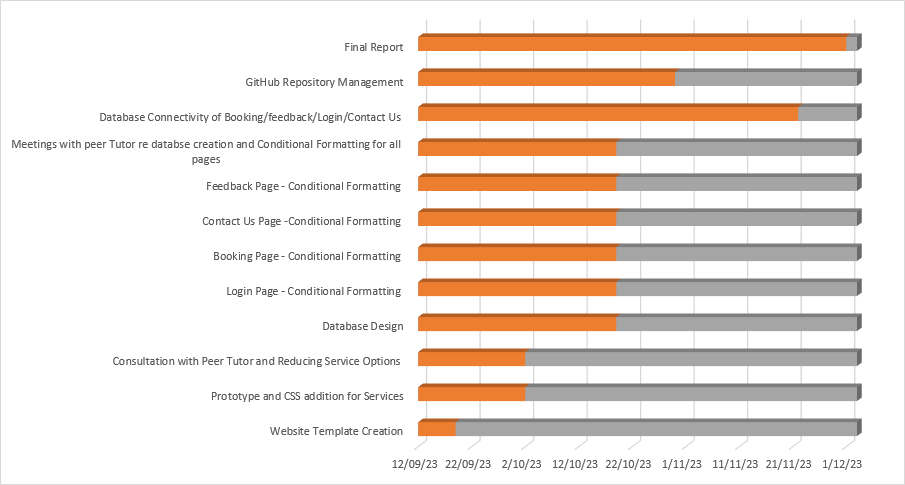
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

11. PROJECT TIMELINE

Having the project timeline served as a constant reminder to understand the roles of our tasks we were committed to completing. The group feel like they achieved the positive result in managing their time effectively, attending regular meetings to assess progress, giving continuous encouragement and being a good resourceful help system when countless bugs were halting our progress.

From week 1 until the end product, we felt like we were constantly in communication and there was high interaction throughout the 3 phases of the timeline, where the simulation of a group environment really help push the team members to finish their tasks productively. Our commitment and dedication towards the end goal of completing the project, would not have been successful without the cohesive unit of the group members working in strong collaboration to do their part and in the end by week 12, we had accomplished the required functionality to make our event website come to life.

| Task | Content | Start Date | End Date | Owner | Duration |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Task-1 | Webiste Template |  |  |  |  |
| 1.1 | Website Template Creation | 12-Sep | 20-Sep | Puneet | 7 |
| 1.2 | Prototype and CSS addition for Services | 26-Sep | 03-Oct | Sachini | 6 |
| 1.3 | Consultation with Peer Tutor and Reducing Service Options | 26-Sep | 03-Oct | Bao | 6 |
|  |  |  |  |  |  |
| Task-2 | Creating Daatabase and CSS Formatting |  |  |  |  |
| 2.1 | Database Design | 03-Oct | 24-Oct | Bao | 16 |
| 2.2 | Login Page - Conditional Formatting | 03-Oct | 24-Oct | Puneet | 16 |
| 2.3 | Booking Page - Conditional Formatting | 03-Oct | 24-Oct | Sachini | 16 |
| 2.4 | Contact Us Page -Conditional Formatting | 03-Oct | 24-Oct | Puneet | 16 |
| 2.5 | Feedback Page - Conditional Formatting | 03-Oct | 24-Oct | Sachini | 16 |
| 2.6 | Meetings with peer Tutor re databse creation and Conditional Formatting for all pages | 03-Oct | 24-Oct | Bao | 16 |
|  |  |  |  |  |  |
| Task-3 | Databse Connectivity and Final Report |  |  |  |  |
| 3.1 | Database Connectivity of Booking/feedback/Login/Contact Us | 24-Oct | 03-Dec | Sachini/Bao/Puneet | 29 |
| 3.2 | GitHub Repository Management | 24-Oct | 31-Oct | Puneet | 6 |
| 3.3 | Final Report | 26-Nov | 03-Dec | Sachini/Bao/Puneet | 5 |



12. ETHICAL AND CULTURAL IMPACT

We think the biggest hurdle affecting not just our software application testing but daily software applications is the strenuous task of making sure the user’s privacy information is not readily available for other users to view in its entirety. Every registered user that has given their personal details has the right to their information being protected from hackers and people who may seriously tamper with their contact details (InfoSecurity Magazine, 2023). This negative action can increase the chance of users receiving unwanted emails and unwanted calls, which is a major roadblock in our company performance in not prioritising our security system to be strong and stable (InfoSecurity Magazine, 2023).

By showing all the users and their bookings in a summary format is an insecure and unsafe way to manage confidential information. Because of modern technological advancement, user’s security is gravely at a risk of being compromised, they may lose trust in our company and cause huge repercussions in spreading negative word of mouth about our company and ruin our event service reputation (InfoSecurity Magazine, 2023).

Another alarming impact is not encrypting the user’s passwords when they register into the website. As hackers become more clever in their efforts to access user personal details, the weak security system we implement related to our password encryption can provide low assurance to our registered users that their login credentials are fully protected and their account will not be bypass (Okta, 2022). The rise in scammers and hackers would give us a serious red flag with our secure password protection, and be prone to more likelihood of malicious cyber attacks of our registered users personal information being exposed to the outside world (Okta, 2022).

13. TEST DOCUMENT

(Need to finalise if edit function works if not remove, test delete by adding contact us info request to show need to cancel bookings)

| **Homepage Navigation Button Test** |  |  |  |
| --- | --- | --- | --- |
| **Choose from the Home, Gallery, Services, Login and Contact Us links in the menu bar, they**  **should direct to the correct pages.** |  |  |  |
| **Result** |  |  |  |
| **All tabs are fully functional, leading to their corresponding pages correctly.** |  |  |  |
| **Registration Form Test** |  |  |  |
| **Go to the login link, click on it to be directed to the login form, at the bottom click on the register link. Fill out register form details and click submit. In the mySQL database check the event users table and get eventusers table page, the added contact details should be present.** |  |  |  |
| **Result** |  |  |  |
| **Everything is working in order, all steps have no issues.** |  |  |  |
| **Login Form Test** |  |  |  |
| **Click the login link to be transfer to the login form. Fill out required details Email: john43@gmail.com Password: J0hn4334. Check to see if details matched in the event users table from mySQL database. Registered user should land on the homepage showing the “Hi, John” message in the top right corner.** |  |  |  |
| **Result** |  |  |  |
| **Failed at first, due to incorrect form code, has been resolve and working fine.** |  |  |  |
| **Form Submission Test** |  |  |  |
| **Using the registration form as an example, fill out the form and press the submit button. If the details have been entered correctly, the data should show up on the mySQL database** |  |  |  |
| **Results** |  |  |  |
| **Due to incorrect coding, failed at first but issue has been resolve, and works fine now.** |  |  |  |
| **Blank Form Test** |  |  |  |
| **Using the registration form, click on the submit button. The result should be there instantly shows a message to fill out required options.** |  |  |  |
| **Result** |  |  |  |
| **Only the first field shows this message, the other fields do not show the required message** |  |  |  |
| **Invalid Login Test** |  |  |  |
| **In the login form, enter the invalid details Email: john43@gmail.com Password: J0hn4443, Session ID will check the details to display an incorrect password or email message to the user.** |  |  |  |
| **Result** |  |  |  |
| **Unsuccessful at first due to incorrect code, issue has been fixed and works fine.** |  |  |  |
| **Examine data is being red from the database** |  |  |  |
| **Open the phpmyAdmin and using example of the geteventusers page, check to see if data is there and all data is being submitted.** |  |  |  |
| **Result** |  |  |  |
| **Unsuccessful at first due to incorrect code, issue has been resolved and works fine.** |  |  |  |
| **Admin Login Test** |  |  |  |
| **On the login page, fill out the valid details in the form Email: admin0108774@hotmail.com Password: 379125admin. The Session ID will look at the registered email and password to check to see if the details match and belong to the admin from the mySQL database then log them to the homepage.** |  |  |  |
| **Results** |  |  |  |
| **Unsuccessful at first due to incorrect code, issue has been resolved and now fully operating.** |  |  |  |
| **Booking Form Test** |  |  |  |
| **After the registered user has logged in, click on the bookings link to be directed to the booking form, fill out details and press submit. The booking information should be added to the database and show the new booking entry.** |  |  |  |
| **Results** |  |  |  |
| **Failed at first, due to incorrect code, still the issue persists in showing only the numerical values of the service options rather than service product names, has not been fully fixed** |  |  |  |
| **Question/Queries Form Test** |  |  |  |
| **For all public users, click on the contact us page and there will be a form to fill out required details and write questions/queries then click submit. The form data should be added to the database, also load the get queries page for admin to see the newly added entry.** |  |  |  |
| **Results** |  |  |  |
| **Failed at first, due to incorrect code in form layout, issue has been fixed and working properly.** |  |  |  |
| **Feedback Form Test** |  |  |  |
| **For registered users, click on the feedback page to fill out a feedback form and click submit. The form data should be added to the database, also admin can load the get feedback page to show the added feedback entry.** |  |  |  |
| **Results** |  |  |  |
| **Works as expected.** |  |  |  |
| **Edit Button Test** |  |  |  |
|  |  |  |  |
| **Results** |  |  |  |
|  |  |  |  |
| **Delete Button Test** |  |  |  |
|  |  |  |  |
| **Results** |  |  |  |
|  |  |  |  |
| **Check that Edit page changes data** |  |  |  |
|  |  |  |  |
| **Results** |  |  |  |
|  |  |  |  |
| **Logout Button Test** |  |  |  |
| **After successfully submitting form, click logout button. The end result should take the user back to the login page.** |  |  |  |
| **Results** |  |  |  |
| **Works as expected.** |  |  |  |

14. USER DOCUMENTATION

For exploring the website, you can navigate across the links at the top of the homepage in showing the main sections of our website. Starting with the eye catching gallery page that have beautiful snapshot images of past events and the services link transporting you to our niche service options with unique and inventive catering choices for food menu packages, cakes and birthday theme settings. Feel free to jump to the contact us page, where you can fill out a questions form to ask any burning questions or queries you may have about our services or anything in general.

There is a login link that is specific for registered users in the pursuit of being able to make an event booking for our services. If you have not created an account, you can click on the register link in the login form to set up an account, therefore you are able to access to the booking page and can follow up with the company to give us feedback.

Regarding the booking form, fill out the details for your event booking and services such as catering, cakes, cake size, text on cake, and birthday theme. Afterwards when all your information is completed, click the submit button and we will get back to you with confirmation and payment instructions by email to notify of your successful booking.

Lastly is the feedback form, where you can type in your email address, select your honest ratings towards our services and give us any suggestions. This feedback form is optional for you after logging in. Whether you choose to provide feedback is entirely up to you, but it would essentially help us improve our service performance. We can get your strong input and process your feedback, letting you know your suggestions are gratefully appreciated. Also we will respond to your suggestions with a follow-up message.

15. CONFIGURATION GUIDE

From the admin’s perspective, open the phpMyAdmin after initialising the Apache and MySQL servers in XAMPP then create your details in the event users table. In the user type field, type in Admin to let the database know you are allocated in this specific role.

Activate the main homepage, head over to the login section and type in your Admin details as it follows email - admin0108774@gmail.com and password - 379125admin to reach your Admin page. Next, browse through the database pages found in the top navigation menu and click “Users” to show all event users information for edit and delete. Follow the same procedure with clicking the bookings to show booking information, queries link showing questions/queries information and feedback showing the feedback form of all registered users.

16. FUTURE SOFTWARE ENHANCEMENTS

There are many add on functionality we can bring to the users and room for improvement to create a more user friendly event website that really digs deep to understand what today’s consumers want when booking an event and scouring different types of services.

At the beginning we had considered having a wide variety of events and services to make a one stop shop for users needing a relevant guide to event booking but in the process of our application it was too much overload of functionality in such a short span of time. The positive benefits of having added services and events broadens the users choices and caters to a larger consumer audience without neglecting potential users who need assistance with their specific events. It also makes our company look more experience in making life a lot easier for all type of users knowing we can provide many types of events and service styles in one place, giving a higher convenience to a wider audience of potential users we are missing out on from condensing our options available.

To solve the key catering problem within the menu section, we would use better stylish names for our menu choices instead of sticking to the default setting. We want the different menus to represent the cultural flavours the customers will be indulging in and make adjustments to the catering field in the booking form to showcase our catering flair helping to build excitement for the event and overall user booking experience.

In the future we would give the registered users the ability to personalise their menus and make the booking experience more interactive. By enabling the registered users to edit or make changes to their current booking, it saves time and energy for the admin having to do all the behind the scenes work to effectively customise the booking in the palm of their hands. It gives them the power to control the booking information and if there are any sudden adjustments, they can promptly send the edit information to the admin, helping dramatically to cut unwanted time and give the admin a faster update response.

For every booking transaction, there needs to be a way for users to pay for their services. Currently, we do not have this payment function and in the future we would make it top priority to show a payment page in order for the registered user to go through with the booking transaction, while providing a more solid event booking experience that feels complete from start to finish. In addition, to keep up to today trends, social media is very essential for the survival of any business therefore for future consideration, we will incorporate social media platforms to our website, where the user can reach us more conveniently and showcase frequent updates to help sustain their interest.

Another area of improvement is the database design of the tables. To optimise the storage of the event users table, the field elements for verified email and active should be change to “Boolean” instead of using “VAR CHAR," as they are solely yes or no response and would save a lot of storage for the table creation TechTarget, 2023). In the database schema, for the event questions table, the name should be changed to questions because it misleads the admin to believe the event questions table only focusses on questions related to events but in reality all types of questions and queries are applicable. Also the cake detail type id table includes the field “cakecategoryid”, this should be changed to cake product since the products table illustrates the cake names rather than the different category of cakes. It is misleading to the admin to give inaccurate information, where the importance of labelling field names in database tables is to prevent confusion in making the database easier to interpret. This will be better implemented in future database table design and show more carefulness in the table construction.

17. EVIDENCE PORTFOLIO

Brendan Cen

The event website project was a group effort, which consisted of 3 members that were assign specific tasks to complete within the given timeframe. For my part, I was responsible for building the ground level structure of the services and booking pages. Although there was clear instructions to work on our sections, the group members were more than responsive to contribute potential ideas for all website pages, assisting in speeding up the process of completing the pages.

In the services page construction, I would do extensive research to look at catering menu options, popular cakes and potential birthday theme through the stock photo websites. Observing event photographers profiles to follow the modernised trends of theme setting was pivotal in making our services have a competitive edge and stay up to date. I would select the most aesthetically appealing images for each service along with inventing the names to complete the services overall character. The 3 main services are in a simplistic table format showing the description followed by colourful images, where the overall mission was to display the services to the customers in a clean and fresh design layout but also promote our niche services effectively. This would potentially drive website traffic, sustain customer’s interest and strengthen the event website appearance.

For the booking page, I had followed similar form layout to the Ganeshan Water Safety app, and altered some field elements to cater to our booking requirements. The connection between the services page and booking page was very close, therefore working on these specific pages enable me to directly take the services main information and add it to the booking form. Completing the booking page served a massive purpose in our website as the registered users were able to book an event with the services, leading to likely increase in booking sales and positive word of mouth to attract new customers in the future. The booking page was the heart of our companies’ website, therefore it provided the platform to allow registered users to choose our finalised services, add further information and ultimately complete the booking procedure.

I was in charge of designing database tables, which was a lengthy process. There were numerous error messages during the creation of the database tables but it was my duty to fix the bugs in order to show the correct data for the admin to view all form data, registered users and booking details. I had created 8 database tables representing the anatomy of the event booking structure, showing the primary and foreign key to link the tables and create relationships (IBM, 2021). The importance of generating the database tables was to design the database schema in accurately displaying the storage of event types, service types, service products, registered users, feedback, questions/queries, and booking details.

From the customers perspective, they can book our events and services while from the companies perspective they can keep tabs of all the registered users information, make required modifications per customers request and store all relevant booking information regarding our niche services in an organised fashion. Also coding the pages for the database using the EJS format to display the database in HTML table for the admin to see the updated data on the website that would greatly benefit the admin in presenting the registered users details, booking information and other form data more clearly and conveniently.

By employing the agile method, I could produce immediate results from my specific tasks in small chunks of the work proportion, and at the same time present my work to the peer tutors and group members to get regular feedback about my stage of progress (Quality Logic, 2023). Improving my work efficiency, time management and strong motivation to make headway with the project objectives. This was a productive method to identify issues earlier on, which enable me to make steady growth in finishing my assigned tasks, that also resonated with the other group members work performance.

The pages I created and database tables played an instrumental part of building the main components of the event website. As the design of the database schema allowed the admin to manage all the registered users information in the best way, not only to keep track of updated information but make necessary adjustments if the customer preference changes. The services and booking page set the foundation for the event website to showcase its most popular feature and sustain the customers’ attention to drive the spotlight for the website.

Sachini Madushani

In the collaborative development of an event management software, I actively contributed to the design and implementation of some pages, with a particular focus on the Gallery page. The Gallery page aimed to provide users with a seamless experience in viewing and interacting with event images. Additionally, I played a key role in addressing and resolving errors that surfaced during the page creation process through consistent communication and collaboration with team members.

The design process involved meticulous planning and execution of the Gallery page, allowing users to effortlessly navigate through a collection of images. Notably, I implemented a feature where users could click on an image to trigger a popup, enabling zoom functionality for a more detailed view. This enhancement aimed to deliver an engaging and interactive experience for users exploring event visuals.

Throughout the development phase, encountering error messages was inevitable. However, our team's commitment to regular meetings proved instrumental in efficiently identifying and resolving these issues. The collaborative problem-solving approach within the team fostered a dynamic environment where challenges were met with collective expertise, resulting in effective solutions.

An integral aspect of the project was linking the designed pages with the database. I actively participated in establishing connections between different pages and the underlying database. This connectivity was crucial for storing and retrieving data accurately, ultimately contributing to the robustness of the software.

Admin Report Generation: As part of the database integration efforts, I played a role in enabling administrators to access a comprehensive final report. This report, derived from the database, provided administrators with a consolidated view of event details. The seamless connection between the front-end pages and the back-end database underscored the importance of accurate and accessible reporting for administrative purposes.

In summary, my contributions to web page design, error resolution, and database integration were pivotal in shaping the success of the event management software. The collaborative approach within the team, coupled with effective communication channels, facilitated a smooth development process despite encountering challenges. The final product reflects our collective dedication to delivering a user-friendly, error-resilient, and database-integrated solution for event management.

18. REFLECTION

Brendan Cen

I feel the immense energy coming together as a team, working on a challenging but exciting project is a good learning curve to overcome continuous obstacles that would be much harder to tackle than working individually. Even though there is high degree of freedom working as solo, the great joys of working in a group environment is the collation of different ideas, sheer enthusiasm and positive encouragement between member keeps me wanting to push forward. Every member contributed with having regular meetings, providing their input on source code production and bouncing off ideas in generating a savvy event website. I felt like I had the group support constantly throughout the software development journey, which was definitely a steep mountain to climb and at times I was completely lost and struggling to understand basic software concepts. It was a good aura knowing the other group members would give me a hand to solve complicated coding and stay laser focus to complete the tasks, where when help was needed I could always count on the group members.

This is entirely new uncharted territory for me, I have never learned any software development material so I was out of my comfort zone but willing to try something to improve myself. Seeing the sharp rise of ChatGPT and AI was a determining factor to do software development. I don’t think my progress would have been possible without the professional support of the amazing peer tutors who have been with me every step of the way like the stakeholders of a company, always being there to give constructive feedback and millions of screenshot of coding error messages, they would rectify the issues and their valuable resources is better than any ChatGPT system.

I definitely found the online classes, much harder than I anticipated to fully understand and the students have been very welcoming to assist the lecturer in helping others sort out problems with their coding, it really felt like they were mini teacher assistants as we are one big happy family supporting each other to succeed. This strong support system has boosted my confidence because we were all in an online environment but working strongly together to achieve a common goal to see everyone make steady progress in their software endeavours.

I have learned the essential need to create user cases, user stories, user case diagram, database schema, ERD diagrams, HTML, CSS, Node.js, MySQL, Javascript, prioritise group tasks and check useful websites such as w3schools and youtube tutorials in enhancing my overall learning, which greatly benefitted me to hone in the basic software development skills. I did feel there was one area, which I had the most problems with being user authorisation incorporating the auth controller pages method from the youtube to assign routes to redirect to the specific user type, where the youtube instructions were fairly complexed and when I try to mirror those coding, I hit a brick wall. Another challenging section was CSS, as I spent a lot of time trying to fix the form positioning rather than focus on basic functionality first. The recommendations from the peer tutor and teachers taught me a valuable lesson to keep things simplistic and not go too overboard with the CSS appearance as the main priority is to establish all required functionality for the website, make sure it is operating, and finally beautify the website pages.

I can really appreciate the wonderful evolving world of software development as an artistic form in creating something truly special and looking back on all the hurdles it was worth the rollercoaster ride that never stops accelerating. If I am to do a similar project, I would incorporate the popular Bootstrap framework to create an industry level website and greatly reduce the timeframe to start the website development from scratch. It has been a pleasure to divulge into something new, knowing the success of ChatGPT and AI will continue to pave the way for future software dominance.

Sachini Madushani

We started to work as a team, and we chose to develop software for an event management company. The project, though complex, became a crucible for collaboration and innovation. Our commitment to open communication fostered an environment where we could freely discuss challenges and collectively devise optimal solutions. Through countless brainstorming sessions and problem-solving discussions, we honed our ability to think critically and work cohesively as a team.

Despite the intricacies of the project, we persevered, and the collective effort paid off. The result was an attractive web application with all the necessary pages seamlessly integrated. Witnessing the transformation of our ideas into a functional product was not only gratifying but also a testament to the effectiveness of teamwork. Each team member played a crucial role, contributing their unique skills and insights to different facets of the project.

The process of developing the web application was not without its share of hurdles. However, these challenges were opportunities for growth and learning. It was during these moments that the strength of our teamwork became most apparent. We learned to leverage each other's strengths, providing support and encouragement when needed, and celebrating together when milestones were achieved.

In the end, the success of our project was not just the web application itself, but the collaborative spirit that fueled its creation. The experience has left an indelible mark on my understanding of teamwork and software development. Finally we develop a product to the event management company. This project has undoubtedly been a cornerstone in my journey as a developer, and the lessons learned will resonate in future endeavors.

Puneet Puneet

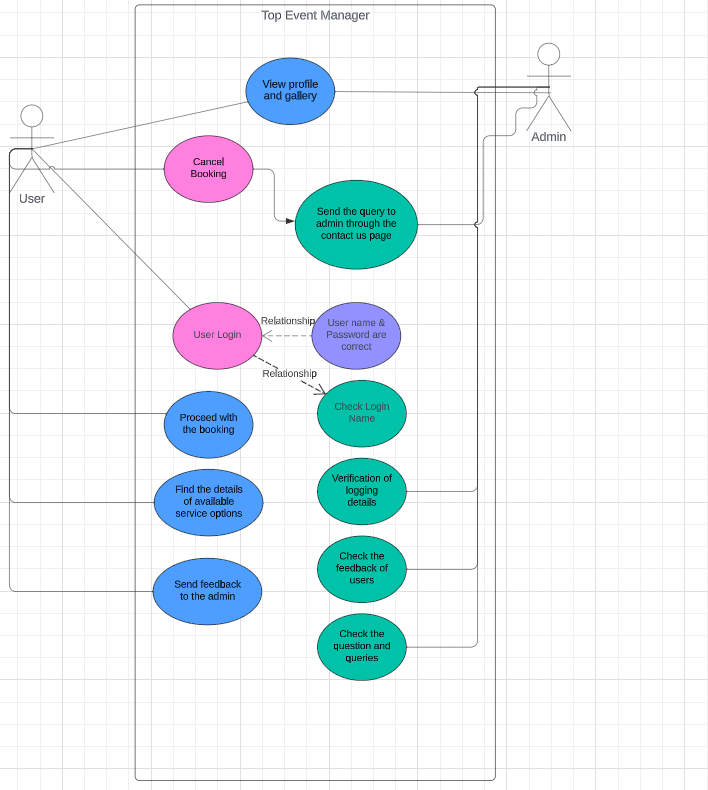
I actively contributed to the design prototype and website template. Website template helped us build the base for the website construction.

The aim was to construct a website that can provide an awesome experience to user.Throughout the development phase, encountering error messages was inevitable. However, our team's commitment to regular meetings proved instrumental in efficiently identifying and resolving these issues. The collaborative problem-solving approach within the team fostered a dynamic environment where challenges were met with collective expertise, resulting in effective solutions.

An integral aspect of the project was linking the designed pages with the database. I actively participated in establishing connections between different pages and the underlying database. This connectivity was crucial for storing and retrieving data accurately, ultimately contributing to the robustness of the software.

In summary, my contributions to web page design, error resolution, and database integration were pivotal in shaping the success of the event management software. The collaborative approach within the team, coupled with effective communication channels, facilitated a smooth development process despite encountering challenges. The final product reflects our collective dedication to delivering a user-friendly, error-resilient, and database-integrated solution for event management.

19. DIAGRAMS



21. REFERENCES 

*Agile Methodology: Meaning advantages, advantages, disadvantages & more.* (2023, March 23). Retrieved from TargetTrend: [*https://targettrend.com/agile-methodology-meaning-advantages-disadvantages-more/*](https://targettrend.com/agile-methodology-meaning-advantages-disadvantages-more/)

*Difference between package.json and package-lock.json.* (n.d.). Retrieved from Medium: *https://umarfarooquekhan.medium.com/difference-between-package-json-and-package-lock-json-6225f83f247c#:~:text=In%20summary%2C%20the%20package.,of%20dependencies%20and%20their%20dependencies.*

*Ethical Hacking: What It Is & Examples.* (2022, June 16). Retrieved from Okta: *https://www.okta.com/identity-101/ethical-hacking/*

*Express Session.* (n.d.). Neanderthals Love Mammals. *https://www.npmjs.com/package/express-session*

*Future Skills Academy, March 2023. HTML/CSS/MySQL.*

*10 Reasons to Use Agile Software Development*. (2023, September 1). Retrieved from *qualitylogic: https://www.qualitylogic.com/knowledge-center/10-reasons-to-use- agile-software-development/*

John Allen, R. H. (2000). EVENTS BEYOND 2000: SETTING THE AGENDA. *AUSTRALIAN CENTRE FOR EVENT MANAGEMENT*, 249.

*Manually starting the DBConfig database configuration tool*. (2023, January 24). Retrieved from *IBM:*<https://www.ibm.com/docs/en/sig-and-i/10.0.0?topic=database-manually-starting-dbconfig-configuration-tool>

*MySQL Left Join Keyword.* (n.d.). Retrieved from w3schools: *https://www.w3schools.com/mysql/mysql\_join\_left.asp*

*New Ethical Concerns in Online Privacy and Data Security.* (2021, February 5). Retrieved from *InfoSecurity Magazine: https://www.infosecurity-magazine.com/next-gen-infosec/ethical-concerns-online-privacy/*

*Primary and Foreign Keys*. (2021, 03 08). Retrieved from *IBM:* [*https://www.ibm.com/docs/en/ida/9.1.1?topic=entities-primary-foreign-keys*](https://www.ibm.com/docs/en/ida/9.1.1?topic=entities-primary-foreign-keys)

*roberthalf. (2023, 09 01). Retrieved froVmie6wBparsoicfiSleDaLnCdMgeatllheorydologies: Which One Is Best?: https://www.roberthalf.com/us/en/insights/career-development/6-basic-sdlc- methodologies-which-one-is-best#One*

*What is Boolean?* (n.d.). Retrieved from TechTarget:<https://www.techtarget.com/whatis/definition/Boolean>

*What is the purpose of the ’node\_modules’ folder?* (n.d.). Retrieved from StackOverflow: *https://stackoverflow.com/questions/63294260/what-is-the-purpose-of-the-node-modules-folder/76049614#76049614*

*What is XAMPP?.* (n.d.). Retrieved from *EDUCBA:*<https://www.educba.com/what-is-xampp/>