WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 3

Applicable VLOs or EESs for This Week's Case Study

4. Evaluate and integrate security features into the client and database application tiers to secure against system threats.

EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 4.7 Analyze, evaluate, and apply relevant information from a variety of sources. (T, A,)
EES 5.9 Interact with others in groups or teams in ways that contribute to effective working relationships and achieving goals. (T, A,)
EES 6.10 Manage the use of time and other resources to complete projects. (T, A,)

This Week's Detailed Case Study Information

This morning, Pitch asks for you and your fellow interns to accompany him to his office. Your team enters his space, a corner office overlooking the dynamic cityscape, and meets eyes with a happy-go-lucky-looking Pitch.

"A sense of pride and responsibility envelopes around our company as it serves millions of users with our extensive library of tunes and seamless playlist creation features," He says immediately upon your arrival, "In alignment with this is our steadfast belief that innovation should be fortified by unwavering security. This belief has prompted me to gather you all here today to embark on a mission that I believe to be both critical and transformative!"

Pitch stands and opens the door to the left of his desk, which leads to a rather large boardroom. Impressed, your team turns inwards to meet each other's wide eyes, eyes that shine with curiosity and anticipation. He invites you into the room to make yourselves comfortable around the long, rectangular table.

"The task in our team's hands this week is profound. You will, with my support, of course, evaluate the existing security measures of the Pockets of Playlists application and database and weave a stronger fabric of protection against lurking threats. We know the digital realm is not just a canvas for creativity but, sadly, is also a battlefield of cyber threats. Our

company's reputation soars high, and this bitterly attracts the attention of malicious actors seeking vulnerabilities to exploit," His voice carrying the weight of urgency.

He elaborates that the company currently has a range of security measures. These include user authentication through password complexity requirements, session management by utilization of secure cookies and ensuring proper session timeouts, and data encryption on sensitive data such as user credentials and payment information in transit. With all of this new information in mind, he commences your team's work for the week.

Deliverables for This Week's Case Study

Your tasks this week include:

- Practice threat modelling and a vulnerability assessment for Pockets of Playlists. You may showcase this in any format your team chooses.
- Develop a presentation about the company's current security measures, including how they provide security against system threats and how these features may be expanded to ensure further security enhancement.
- Create an educational video about security testing tools and how they may be used with Pockets of Playlists.
- Write a blog about the difference in security features in the client versus the database application tiers.

Pockets Of Playlists

(CSAT-3)

Arvind Singh jugtwan (C0860886)

NagaRaju Tallapelly (C0859913)

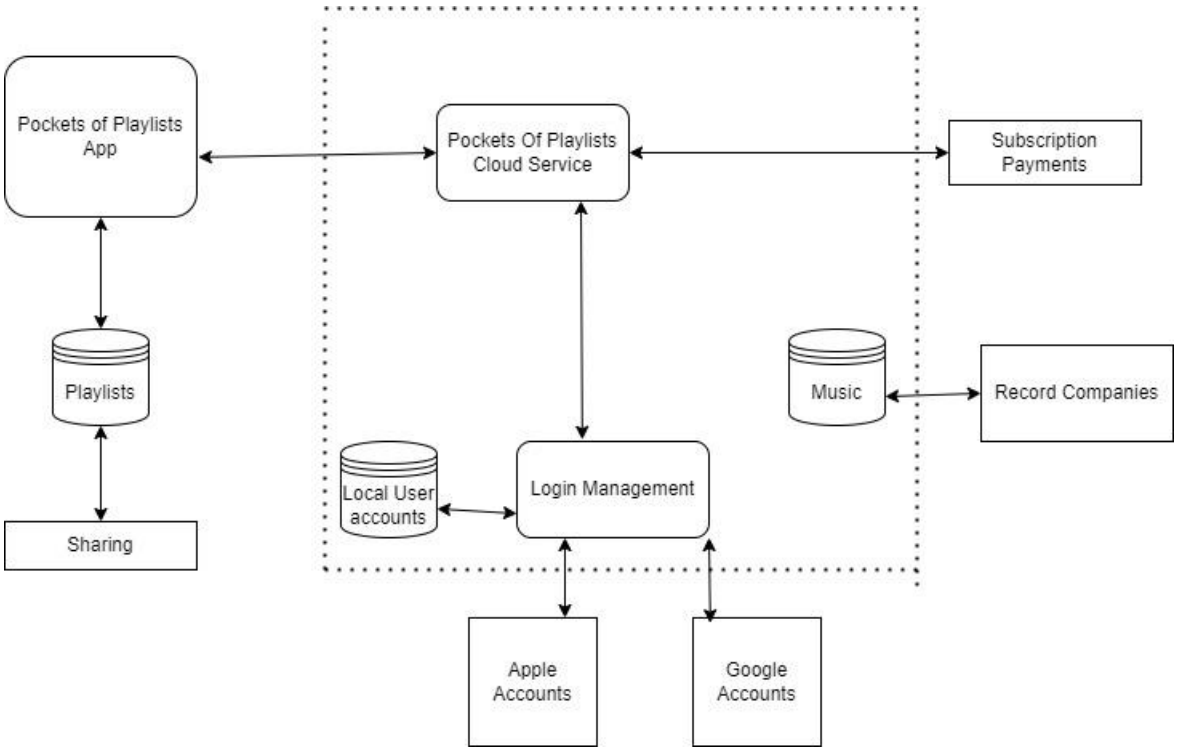
Rahul Ravindra Madeshiya (C0860488)

Amrutha Jayanthimala Sukumaran(C0860921)

Introduction

In the contemporary landscape, the significance of application security within the broader realm of network security cannot be overstated. In a world where hackers continually employ new technologies and tactics to gain unauthorized access to crucial data and engage in malicious activities, safeguarding applications and the valuable information they contain has become paramount. Regrettably, many have encountered significant setbacks in their approach to application security, leading to the oversight of numerous vulnerabilities and subsequent attacks that have inflicted harm. Consequently, the importance of both network and application security has never been more pronounced in today's interconnected world. Threat modeling serves as a crucial process to enhance network security, involving the delineation of objectives and vulnerabilities. This information is instrumental in understanding the motivations and techniques that potential attackers might employ to exploit vulnerabilities or pose threats to a system.

THREAT MODELLING DIAGRAM



Threat Modelling:

1. Assets : The Main Assets of Pockets of Playlists are User Accounts, Subscription Payments, Playlists and music or podcasts created by users.
2. Threats : Pockets of Playlists has potential Threats such as data breach, bank information leak while subscription payments , users playlists, artists profiles.

Vulnerability Assessment :

A vulnerability assessment is a methodical examination of security weaknesses within an information system. It involves determining whether the system is vulnerable to known security flaws, assigning severity ratings to these vulnerabilities, and recommending actions for remediation or mitigation when necessary.

Vulnerability assessments can help prevent various threats, such as:

1. Attacks involving the injection of code, like SQL injection and cross-site scripting (XSS).

2. Unauthorized privilege escalation resulting from flawed authentication methods.

3. Security risks associated with software that comes with insecure configurations, such as easily guessable admin passwords.

There are various types of vulnerability assessments:

1. Host assessment: This involves evaluating critical servers that may be susceptible to attacks if they haven't been adequately tested or if they were not created from a secure machine image.

2. Network and wireless assessment: This assesses policies and practices put in place to prevent unauthorized access to both private and public networks, as well as resources accessible through these networks.

3. Database assessment: This focuses on identifying vulnerabilities and misconfigurations within databases or big data systems. It also involves identifying unauthorized databases or insecure development/testing environments and classifying sensitive data throughout an organization's infrastructure.

4. Application scans: This entails the detection of security vulnerabilities in web applications and their source code. It is achieved through automated scans of the front-end or through static and dynamic analysis of the source code.

Blog about the difference in security features in the client versus the database application tiers.

In an era where digital data has become the lifeblood of businesses and individuals alike, security is an ever-present concern. A variety of approaches involving the implementation of security measures at multiple layers of the technological stack is needed to safeguard sensitive data and guarantee the integrity of applications. The client and database application tiers are two crucial layers in this security architecture, and they each play a crucial role in protecting data and systems. But they deal with quite distinct security issues, and they use very different solutions.

The goal of this blog is to examine the variations in security features between the client and database application layers. By being aware of these differences, we can more clearly understand the complexity of contemporary security tactics and how they cooperate to defend against a variety of threats. Come explore with us the distinct features and difficulties of each tier's.

The client Tier :

User Authentication :

1. Mechanisms for Authentication

User authentication is often the first step in client-side security. At this level, password-based authentication, biometrics, or multi-factor authentication (MFA) are the available authentication methods. Here, the emphasis is on confirming users' identities before allowing them access to the program.

2. **User Control Access:**

Role-based access control (RBAC) and permissions management are prominent features on the client tier. 2. User Access Control. These controls guarantee that users can only access the application's sections that are pertinent to their roles and responsibilities.

Data Encryption :

1. **Data in Transit:** When data moves through the network, it must be secured while in transit. Data transmission between the client and server is frequently encrypted using the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.
2. **Data at Rest**

Client device data that is kept locally, such as on laptops or mobile devices, should be encrypted. In the event that the device is lost or stolen, this prevents unauthorized access to important information.

The Database Application Tier :

Access Control:

1. **Database access controls** : These are managed by the database application tier under access control one. User accounts and privileges are used by database management systems (DBMS) to limit who can view, write, and modify data. To prevent unwanted access, access control methods must be properly implemented.
2. **Application-level Access Control**: The application layer implements its own access controls in addition to database access. By guaranteeing that only authorized users can carry out particular tasks within the application, this offers an additional degree of protection.

Data Encryption :

1. **Data Encryption at Rest:** Databases house a substantial volume of private data. Data must be encrypted while it is at rest in order to prevent theft or unwanted access. For this purpose, a lot of DBMS solutions come with built-in encryption features.

2. **Data masking**: In some circumstances, sensitive data can be disguised so that users who don't require to view the complete information are not made aware of it. For apps that deal with sensitive personal or financial data, this is especially crucial.

Video Link :

https://mylambton-my.sharepoint.com/:v:/g/personal/c0860488_mylambton_ca/EUHaoPwNq_BJiRSct1xMzYYBOdazQAcYU-5KthY8R8LnuQ?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0RpcmVjdCJ9fQ&e=iheF6u

Conclusion :

In conclusion, the client and database application tiers' security features have different but connected functions. User authentication, access management, and data encryption in transit and at rest are the main areas of concentration for the client tier. In contrast, the database application tier places a strong emphasis on granular access control, auditing, backup, and recovery.

Given that vulnerabilities can affect system security at any level, a comprehensive security strategy must include both tiers. In order to protect sensitive data and preserve user and customer confidence, it is essential to stay up to date on new security risks and best practices for client and database applications.

References:

1. *Shostack + Friends Blog > Threat Modeling Open training: First quarter, 2022*. (n.d.). https://shostack.org/blog/threat-modeling-open-training/

2. *Threat Modeling 101: Getting started with application security threat modeling [2021 update] | Infosec*. (n.d.). https://resources.infosecinstitute.com/topics/management-compliance-auditing/applications-threat-modeling/
3. Richardson, L. (2023, March 22). *What is Vulnerability Assessment | VA Tools and Best Practices | Imperva*. Learning Center. https://www.imperva.com/learn/application-security/vulnerability-assessment/

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 4

Applicable VLOs or EESs for This Week's Case Study

5. Integrate software applications for teams to collaborate and control the outcomes of a project.

EES 3.4 Apply a systematic approach to solve problems. (T, A,)
EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 4.6 Locate, select, organize, and document information using appropriate technology and information systems. (T, A,)
EES 6.11 Take responsibility for one's actions, decisions, and consequences. (T, A,)

This Week's Detailed Case Study Information

"Good Morning, friends!" The pitch says as he side slides into your team's workspace, his animated behaviour making you and your peers giggle, "Let us begin your third week, shall we? As you know, our goal here at Pockets of Playlists is to offer users a realm of songs at their fingertips and a platform to curate personalized playlists. Lately, we have begun to engage in conversations about how we envision elevating our company's offerings to greater heights!"

He explains that as the demand for new features and enhanced user experiences grows, Pockets of Playlists' team has equally grown a recognition of the pivotal role of effective collaboration among its diverse development teams.

"Of course, we are no stranger to the concept of collaboration, as your team will witness next week and moving forward through our weekly Monday meetings where stakeholders from various corners of the organization convene to discuss progress. Yet, I come forth with this conversation today as, despite these well-intentioned efforts, challenges in fostering a unified vision and controlling project outcomes have persisted throughout our company," He says, his energy lower than you've seen it in the past.

"A realization has struck me that it is time for a transformative shift. So, without further ado, I call upon your budding minds to assist me in orchestrating the integration of software

applications to bind the development teams in a collaborative tapestry that will transcend boundaries."

As Pitch concluded his rather dramatic monologue, his vision was clear: to foster an environment where cross-functional teams seamlessly communicated, collaborated, and contributed to the shared goals of innovation and enhanced user experiences. Through such, the Pockets of Playlists app could transcend its status quo to embrace a new era of collaborative brilliance.

Deliverables for This Week's Case Study

Your tasks this week include:

- Evaluate and select communication tools such as Slack, Microsoft Teams, or other platforms for real-time messaging and team collaboration.
- Review and choose a project management software to track tasks, assign responsibilities, and monitor progress.
- Write a newsletter about your chosen communication tools and project management software to educate and inform the employees of Pockets of Playlists.
- Contact at least three real-life industry professionals to gain their insight about software applications to integrate to ensure collaboration among teams and help control project outcomes.
- Portray your team's learnings from real-life industry professionals in any creative format you wish.

COMMUNICATION
AND
PROJECT MANAGEMENT TOOLS

CSAT-3 (WEEK-4)

Team Leader: RAHUL RAVINDRA MADESHIYA

TEAM MEMBERS

Rahul Ravindra Madeshiya (C0860488)
Nagaraju Tallapelly (C0859913)
Amrutha Jayanthimala Sukumaran (C0860921)
Arvind Singh Jugtwan (C0860886)

1. COMMUNICATION TOOL FOR REAL-TIME MESSAGING AND TEAM COLLABORATION

SLACK APPLICATION

Slack is a messaging platform primarily developed for corporate usage, aimed at simplifying the exchange of information and fostering teamwork among team members. It stands at the forefront of reshaping how businesses communicate by emphasizing creating a seamless work environment and enhancing collaborative efforts.

Slack enhances our work environment by fostering connectivity, adaptability, and inclusivity.

- **Connected**
  Slack simplifies the process of connecting with colleagues, whether they are part of your organization or external partners, facilitating seamless collaboration reminiscent of in-person interactions. It organizes work into specialized spaces called channels, which help bring together the right people and relevant information.

- **Flexible**
  Slack supports flexible and asynchronous work approaches. With information neatly structured within channels, factors like your location, time zone, or role become less critical. You can access the necessary information at your convenience, ask questions, stay updated, and share updates without the need for scheduling coordination.

- **Inclusive**
  Within Slack, every member of your organization has equal access to shared and searchable information. Collaborative work in channels ensures that information is disseminated to everyone simultaneously, promoting team cohesion and expediting decision-making processes.

Features of the SLACK Application

Channels - Slack utilizes channels to structure discussions, enabling the creation of channels for specific teams, projects, or topics, thereby facilitating communication and collaboration among team members within these dedicated spaces.

Direct Messaging - Users could engage in private one-on-one conversations through direct messaging with their colleagues.

File Sharing - Slack allows the sharing of various types of content, including files, documents, and images, within the team. It is also seamlessly integrated with various file-sharing services.

Search Functionality - Slack featured a robust search function that simplified the process of finding specific messages, files, or conversations, even within extensive archives.

Integrations - Slack seamlessly integrated with a wide range of third-party applications and services, such as Google Drive, Dropbox, Trello, and others, streamlining workflow and data sharing.

Notifications - Users had the ability to customize their notification settings, enabling them to receive alerts for specific keywords, mentions, or channels, ensuring they stayed informed without feeling overwhelmed.

Threaded Conversations - Slack allowed the creation of threaded conversations within channels, enhancing the ability to track and participate in specific discussions.

Video and Voice Calls - Slack provides video and voice calling features suitable for both one-on-one and group conversations.

Channel Mentioning - Users could mention specific channels or team members in messages to draw attention to important information.

Message Formatting - Slack-supported message formatting options, including bold, italics, and code blocks, making messages more readable and structured.

Analytics and Insights - Slack provided analytics and insights into team usage, assisting administrators in monitoring engagement and activity levels.

Security - Slack placed a strong emphasis on security, implementing features such as two-factor authentication, data encryption, and compliance standards to safeguard user information.

2. PROJECT MANAGEMENT SOFTWARE

JIRA APPLICATION

Jira is a software application created by the Australian company Atlassian, with the purpose of enabling teams to track issues, manage projects, and optimize workflows.

Jira software features are as follows:

Issue Monitoring - Jira excels in the realm of tracking issues, empowering users to create, assign, prioritize, and trace issues or tasks across the project's lifecycle.

Project Oversight - Jira equips users with tools for planning projects, managing tasks, and overseeing project progression, ensuring teams remain well-organized and adhere to schedules.

Tailored Workflows - Jira allows users to craft customized workflows that align precisely with their distinct processes and business prerequisites, rendering it highly adaptable for various project types.

Support for Agile Approaches - Jira extends support to various agile methodologies, including Scrum and Kanban, featuring functionalities like sprint planning, backlog management, and the creation of burndown charts.

Reporting and Dashboard Creation - Users can produce reports and establish personalized dashboards for visualizing project advancement, tracking crucial metrics, and gaining insights into team performance.

3. NEWSLETTER: COMMUNICATION AND PROJECT MANAGEMENT TOOL

Hello Pockets of Playlist team, we have selected two tools for our application.

1. SLACK (Communication Tool)
2. JIRA (Project Management Tool)

As we know, in this current evolving world communication within the team is an important aspect of developing an error-free application so we have selected SLACK. For managing the project status, we are using JIRA.

Why are we using JIRA and SLCAK in our application?

Our main goal as a pocket of the playlist team is to develop a smooth experience for the end-users who will be using this application for music consumption. To improve our project-related task, it is very important to keep a detailed eye on every minute of errors and also we need to meet the project deadline. JIRA provides an extensive plugin and real-time tracking of the application's progress.

Similarly, SLACK provides tons of key features for accurate communication among team members. The features are as follows, File Sharing, Direct Messaging, Notifications, Audio and Video Calls, and Security of the confidential data.

SLACK AND JIRA USED IN SOME IMPORTANT PROJECT DEVELOPMENTS

SLACK LOGO

- Workplace Communication
- Single Platform for messaging, tools, and file sharing.
- AI tools used for automation.

JIRA LOGO

- Work Management
- IT Service Management
- Agile and DevOps
- JIRA Tickets

4. THREE REAL-LIFE INDUSTRY PROFESSIONALS KNOWLEDGE

We have communicated with real-life industry professionals and their names are below.

- Vikash Samota (Technical Led – Web Development)
- Arvind Singh Jugtwan (Full Stack Developer)
- Rahul Madeshiya (Spatial Data Specialist)

Effective collaboration among teams and maintaining control over projects can be achieved through the integration of diverse software applications and tools.

Tools for Managing Projects - Trello, Asana, JIRA
Communication and Messaging - Slack, Microsoft Teams
File Sharing and Storage - Google Drive, Dropbox
Collaborative Document Handling - Microsoft SharePoint, Confluence
Task Automation - Zapier, Integromat
Visual Collaboration - Miro, Lucidspark
Data Analytics and Reporting - Tableau, Power BI
Human Resources and Team Management - BambooHR, Workday

5. TEAM LEARNINGS

Our team has learned multiple things about how to manage the products using various software applications. In today's world, communication application plays an important role in connecting people around the world. As a team, our focus should be on developing and maintaining the application, fixing the bugs and making it easy to use for the people.

There are lots of software applications in the app store. But we need to choose the application which fits our requirements. It is an important aspect of software development to get 100% from the software application.

REFERENCES

Streamline your cloud operations with Slack and AWS Chatbot. (n.d.). https://d34u8crftukxnk.cloudfront.net/slackpress/prod/sites/6/Slack-AWS-ebook-FINAL.pdf

Slack. (n.d.). *What is Slack?*. Slack Help Center. https://slack.com/help/articles/115004071768-What-is-Slack-

Slack. (n.d.-b). *Where work happens*. Slack Help Center. https://slack.com/help/categories/200111606

Logo Library - Resources. Atlassian Design System. (n.d.). https://atlassian.design/resources/logo-library/

Jira. What is Jira? | Definition and Overview. (2022, November 30). https://www.productplan.com/glossary/jira/

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 5

Applicable VLOs or EESs for This Week's Case Study

2. Design, model, implement, maintain, and query databases using an enterprise-level relational database management system (DBMS) to satisfy end-user specifications.

EES 1.1 Communicate clearly, concisely, and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience. (T, A,)
EES 3.4 Apply a systematic approach to solve problems. (T, A,)
EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 4.6 Locate, select, organize, and document information using appropriate technology and information systems. (T, A,)

This Week's Detailed Case Study Information

Your week begins differently than in the past few weeks at Pockets of Playlists. Rather than individual meetings with Pitch and your team, you and your peers are set to meet with the entire team of Pockets of Playlists for their regularly scheduled Monday morning meeting. At the beginning of last week, Pitch informed your team the purpose of these meetings was to ensure collaboration among the company. As the day has come, you are overly excited to witness this process.

As you enter the boardroom, you are met with an array of smiles on the faces of many Pockets of Playlists professionals. You smile in response and plop down into the closest chair around the table. Pitch stands at the head of the table, bouncing on the balls of his feet.

"Let us get this meeting started, shall we?" He voices after the final chair is occupied, "The urgency to refine our company's underlying database has recently come to my attention. I understood this as I pondered our company's drive to offer our users an unparalleled user experience."

Pitch continues to explain his vision of a database that harmonized seamlessly with the app's functionality, ensuring every user's journey was smooth and melodious. He stated that while the database was functional, its optimization was paramount to delivering the app's promise.

He then presented to the group a set of end-user specifications to guide the team's journey this week. This set included users' expectations of near-instantaneous playlist loading times, swift song retrievals, and an app experience as harmonious as a perfectly-composed symphony.

After allowing chatter, Pitch pulled in the conversational reigns again and directed his attention towards your group. He shared that your mission is to join the budding software and database enthusiasts' team in optimizing Pockets of Playlists' existing database management system. He elaborates that by designing, modelling, implementing, maintaining, and skillfully querying the database, your team can ensure that the enterprise-level relational DBMS aligns with end-user specifications, thus delivering a seamless and delightful musical experience.

"The database is the backbone of Pockets of Playlists," Pitch emphasized, "It's not just about storing data; it's about delivering the enchantment of music to our users. Remember this as you all embark on your work for the week."

Deliverables for This Week's Case Study

Your tasks this week include:

- Select a subset of Pockets of Playlists' entities (e.g., songs, playlists, users) and design an entity-relationship diagram for them.
- Construct a list of end-user specifications, including the examples offered by your supervisor and additional ideas from your team.
- Develop a presentation to explain your team's database optimization choices to your supervisor.
- Record your team presenting your strategies to your supervisor. Be engaging and creative!

CASE STUDY

CSAT-3 (WEEK-5)

Team Leader: NAGARAJU TALLAPELLY

TEAM MEMBERS

Rahul Ravindra Madeshiya (C0860488)

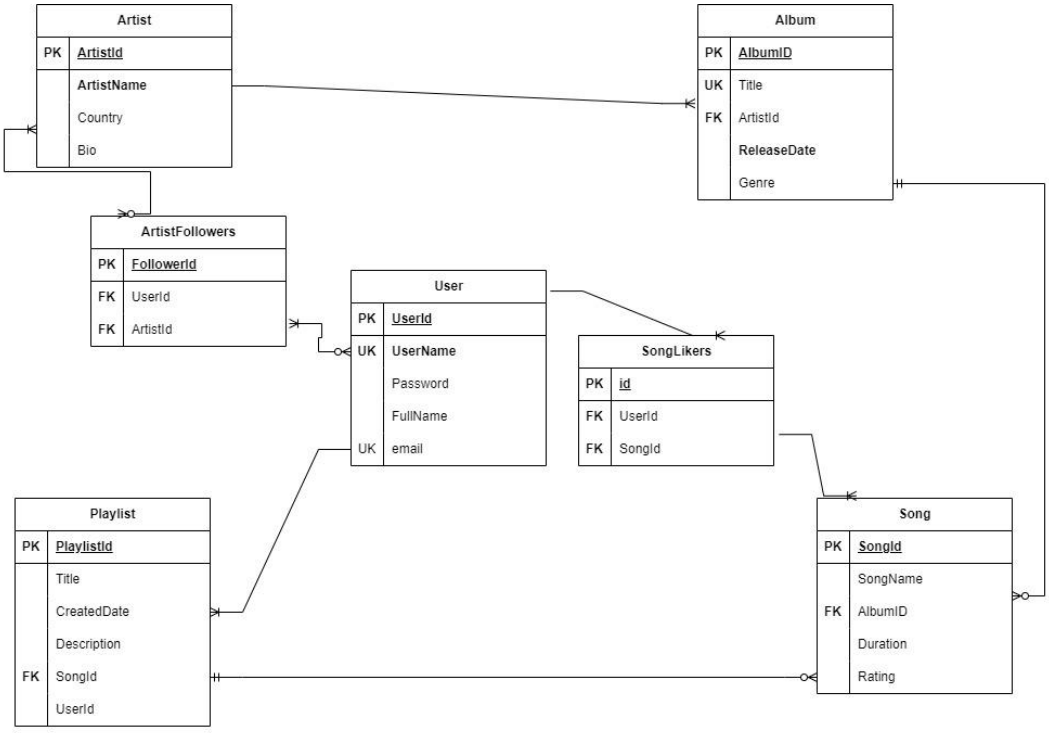
Nagaraju Tallapelly (C0859913)

Amrutha Jayanthimala Sukumaran (C0860921)

Arvind Singh Jugtwan (C0860886)

1. ENTITY-RELATIONSHIP DIAGRAM FOR POCKETS OF PLAYLIST (APPLICATION)

ER-DIAGRAM: POCKETS OF PLAYLIST ER-DATABASE MODEL



In this ER Diagram, we have created 7 database tables,

1. **Artist:** (Aristid, ArtistName, Country, Bio)
2. **ArtistFollowers:** (FollowedId, UserId, ArtistId)
3. **Playlist:** (Playlistid, Title, CreateDate, Description, SongId, UserId)
4. **Users:** (UserId, UserName, Password, FullName, Email)
5. **SongLikers:** (Id, UserId, SongId)
6. **Album:** (AlbumId, Title, ArtistId, Release Date, Genre)
7. **Song:** (SongId, SongName, AlbumId, Duration, Rating)

2. **A list of end-user specifications for a music playlist app involves understanding the features and functionalities that users would expect.**

Below is a comprehensive list of end-user specifications for a music playlist app, including examples and additional ideas:

Profile management and user registration :

- User accounts can be created with unique usernames and passwords.
- With the use of display names, bios, and avatars, users can customize their profiles.
- Social media login options for speedy registration and profile linkage.

Music Library Integration:

- The app allows users to import music from their devices or online storage.

Playlist Creation and Management:

- Users can create multiple playlists with unique names.
- Rearrange, add, or remove music from playlists.

Search and Discovery:

- A powerful search engine to find albums, artists, and songs.
- Music suggestions that are unique to you based on your listening habits.

Offline Mode:

- Being able to download music for offline listening.

Social and Sharing:

- Share playlists, songs, and favourite tracks with friends and followers.
- Like and comment on tracks or playlists.

Privacy and Security:

- You can use privacy settings to limit who can access and engage with your profile.
- Secure login methods and data encryption.

Cross-Platform Compatibility:

- Availability across multiple operating systems, including web, iOS, and Android.

3. POWER-POINT PRESENTATION FOR TEAMS DATABASE OPTIMIZATION

ONEDRIVE LINK:

https://mylambton-my.sharepoint.com/:p:/g/personal/c0860488_mylambton_ca/EWm5AJi2cDpNhvp-PWjEP-EBG0NbLx0KrM5lwSrTn8264w?e=T4L180

4. TEAM PRESENTATION VIDEO LINK

ONEDRIVE LINK:

https://mylambton-my.sharepoint.com/personal/c0860488_mylambton_ca/_layouts/15/stream.aspx?id=%2Fpersonal%2Fc0860488%5Fmylambton%5Fca%2FDocuments%2FCSAT%2D3%20%28WEEK%2D5%29%20FILES%2FCSAT%2D3%20%28Week%2D5%29%20Team%20Discussion%20%20Video%2Emp4&nav=eyJy

REFERENCES

Web API | Spotify for Developers. (n.d.). https://developer.spotify.com/documentation/web-api

Systems, O. (n.d.). *User Requirement Specifications (User Specs, URS) | OFNI Systems*. Ofni Systems. https://www.ofnisystems.com/services/validation/user-requirement-specifications/

Lane, G. K. C. (2023). How to write a Software Requirements Specification (SRS document). *Perforce Software*. https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 6

Applicable VLOs or EESs for This Week's Case Study

1. Evaluate system requirements and implement multi-tiered (client, server, and database) web applications to meet client requirements.

EES 2.3 Execute mathematical operations accurately. (T, A,)
EES 3.4 Apply a systematic approach to solve problems. (T, A,)
EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 4.7 Analyze, evaluate, and apply relevant information from a variety of sources. (T, A,)

This Week's Detailed Case Study Information

It's today's Monday morning meeting; you learned that the company yearns to extend its digital reach and weave its melodies into the very fabric of the online web. Pitch shared he would like every member's help to lead this journey from the client tier, the server tier, to the backbone, the database tier.

"As we begin our journey with the client tier, where the magic of the user interface is unfolded, we will create an interface that resonates with the familiar aesthetics of our company's mobile app. It will be user-friendly, responsive, and warm and welcoming, no matter what devices our friends may use," Pitch voices.

You're caught off guard as one of the team members you met the previous week, Martin, stands to his feet to join Pitch. Pitch smiles at Martin and motions his hands his way.

"Yes, Pitch has asked that I discuss the remaining tiers given my role as a full-stack developer with the company. Quite frankly, what I would like to emphasize most here is scalability and performance. We aim to develop a backend that handles user authentication, song recommendations, and playlist management with finesse."

Martin continues to elaborate on the "backbone" or, rather, the database tier. He discusses that in this very tier, the team must carefully craft the architecture that stores user profiles,

songs, playlists, and interactions. During this, data integrity and responsiveness must be ensured.

As the meeting was coming to be adjourned, Pitch reminded the team of the goal at hand. He states that within the next week, the group, including your group, will create a multi-tiered web application that seamlessly echoes the offerings of its acclaimed mobile app.

"This new symphony will be a digital symposium, allowing users to access their favourite tunes across different devices and platforms!"

Deliverables for This Week's Case Study

Your task this week includes:

- Record and roleplay your team, collaborating with stakeholders to gather comprehensive system requirements and understand the desired functionalities and user interactions.
- Write a summary of your team's gained understanding due to your collaboration.
- Identify key features such as user authentication, song browsing, playlist creation, and sharing.
- Divide your group into two smaller groups.
- Group one will write a step-by-step outline about designing and developing the client-side interface using modern web technologies to create a visually appealing and responsive user interface.
- Group two will write a step-by-step outline on server-side development, including a summary of the purpose for each step chosen.

WIL PROJECT
(CASE STUDY)

CSAT-3 (WEEK 6)

CSAT-3 (WEEK-6)
Team Leader: ARVIND SINGH JUGTWAN

TEAM MEMBERS
Rahul Ravindra Madeshiya (C0860488)
Nagaraju Tallapelly (C0859913)
Amrutha Jayanthimala Sukumaran (C0860921)
Arvind Singh Jugtwan (C0860886)

1. **Roleplay of our team collaborating with stakeholders to gather system requirements and understand the desired functionalities and user interactions.**

   Roleplay characters are as follows.

ROLE PLAY CHARACTER	NAME
Stakeholder	Amrutha Jayanthimala Sukumaran
Team Leader	Arvind Singh Jugtwan
Software Developer	Nagaraju Tallapelly
Application Tester	Rahul Ravindra Madeshiya

   Video Link: https://mylambton-my.sharepoint.com/

2. **Summary of our team gained understanding due to the collaboration.**

   Collaborating with stakeholders has had a substantial impact on our team, leading to a more profound understanding and improved project outcomes. Here's a summary of the key benefits:

   Contextual Awareness: Our collaboration with stakeholders has broadened our understanding of the project's context and objectives. This enhanced context enables us to make more informed decisions and align our efforts effectively.

   User-Centred Approach: Engaging with stakeholders has provided valuable insights into the needs and preferences of end-users. This understanding has empowered us to tailor our solutions to match user expectations better.

   Realistic Planning: Stakeholder involvement has illuminated practical constraints, including budget limitations, regulatory requirements, and time constraints. Recognizing and addressing these constraints has allowed us to create more achievable and realistic project plans.

   Expert Insights: Collaborating with stakeholders has deepened our knowledge in complex areas. Their guidance has been instrumental in resolving technical challenges and making well-informed design decisions.

   Continuous Improvement: Regular interaction with stakeholders has established feedback loops, fostering an iterative approach to problem-solving. This agility has led to ongoing enhancements in our work.

   Risk Mitigation: Stakeholders have identified potential risks and vulnerabilities, enabling us to proactively address these issues and establish contingency plans. This proactive approach reduces project disruptions.

Alignment with Goals: Through collaboration, we have ensured that our team's efforts closely align with the overarching project or organizational objectives. This alignment has improved overall efficiency and effectiveness.

Adaptability: The dynamic nature of stakeholder input has made our team more adaptable. We've learned to pivot and adjust our strategies and solutions to accommodate evolving requirements and changing priorities.

In summary, collaborating with stakeholders has significantly enriched our team's understanding in various dimensions, including project context, user needs, and technical intricacies. This collaborative approach has not only enhanced the quality of our work but has also cultivated a more versatile and resilient team capable of navigating the complexities of demanding projects effectively.

3. **Identifying key features such as user authentication, song browsing, playlist creation, and sharing.**

Identifying critical elements in a project, such as in the context of a music streaming application, is pivotal for both successful development and meeting user expectations. Here are some of the fundamental characteristics:

User Verification: User Registration: Allowing individuals to create accounts by inputting their personal information. Login: Offering a secure method for users to access their accounts using authentication techniques such as passwords or biometrics. Password Recovery: Enabling users to reset their passwords if they forget them.

Song Exploration: Search Functionality: Enabling users to search for specific songs, artists, albums, or genres. Category-Based Navigation: Organizing songs into categories based on genres, artists, and albums to simplify the discovery process. Personalized Song Suggestions: Supplying song recommendations tailored to individual user preferences and listening history.

Playlist Control: Playlist Creation: Empower users to create custom playlists containing their favourite tracks. Adding/Removing Songs: Allowing users to add or remove songs from their playlists. Playlist Management: Providing options for editing, reordering, and deleting playlists.

Content Sharing: Social Media Sharing: Allowing users to share songs or playlists on their social media platforms. Collaborative Playlist Building: Enabling multiple users to collaboratively create and modify playlists. Shareable Links: Furnishing unique links for sharing playlists with non-users. These fundamental features are crucial for ensuring a seamless and enjoyable user experience within a music streaming app.

4. **Divide our group into two smaller groups.**

GROUP-1 (CLIENT-SIDE)	GROUP-2 (SERVER-SIDE)
Rahul Ravindra Madeshiya	Nagaraju Tallapelly
Amrutha Jayanthimala Sukumaran	Arvind Singh Jugtwan

5. **Group 1, step-by-step outline about designing the client-side interface using modern web technologies to create a visually appealing and responsive user interface.**

Step 1: Defining Objectives and User Requirements
Start by outlining the project's goals, understanding the target audience, and identifying the essential user needs. Determine the core components and content that should take center stage within the interface.

Step 2: Conceptualization and Sketching the Design
Begin by creating initial sketches and wireframes to visualize the layout and arrangement. Pay close attention to the placement of vital elements, such as navigation, content, and interactive features.

Step 3: Selecting the Right Technologies
Choose the appropriate modern web technologies for our project, which could include HTML5, CSS3, and JavaScript frameworks like React, Angular, or Vue.js.

Step 4: Ensuring Responsiveness
Apply responsive design principles to ensure the interface seamlessly adapts to various screen sizes, including desktop, tablet, and mobile. Utilize media queries and adopt a mobile-first approach for optimal responsiveness.

Step 5: Designing the User Interface
Create an aesthetically pleasing user interface by carefully selecting a colour scheme, typography, and visuals that align with the project's identity and user preferences. Prioritize user accessibility to ensure text and images are legible and navigation is user-friendly.
Step 6: Styling with CSS
Develop well-structured CSS code to style the interface. Consider using CSS preprocessors like SASS or LESS for more efficient development. Explore the use of CSS frameworks such as Bootstrap or Material-UI for streamlined styling.

Step 7: Implementing Interactivity
Introduce interactive elements, including buttons, forms, and navigation menus. Utilize JavaScript to add dynamic behaviours and engaging animations. Ensure that transitions and animations are seamless, contributing to an immersive user experience.

Step 8: Optimizing Performance
Optimize assets like images to reduce load times. Minify and compress CSS and JavaScript files to enhance page rendering speed. Employ strategies like lazy loading and asynchronous loading for resources.

Step 9: Ensuring Cross-Browser Compatibility
Rigorously test the interface in various web browsers, including Chrome, Firefox, Safari, and Edge, to ensure consistent performance and appearance. Address any browser-specific issues by adjusting CSS or JavaScript.

Step 10: Prioritizing Accessibility
Guarantee that the interface is accessible to all users, including those with disabilities. Utilize semantic HTML elements, provide descriptive alt text for images, and enable keyboard navigation.

Step 11: Conducting User Testing
Engage in usability tests with real users to gather feedback on the functionality and design of the interface. Make necessary refinements based on user insights.

Step 12: Optimizing for Mobile
Fine-tune the mobile experience, considering touch gestures, limited screen space, and optimized performance on mobile devices.

Step 13: Ensuring Security
Implement robust security practices, including input validation and safeguards against common web vulnerabilities like XSS and CSRF.

Step 14: Documentation
Document your code, design choices, and the chosen technology stack for future reference and collaboration with the team.

Step 15: Continuous Improvement
Regularly update and maintain the interface to stay aligned with evolving web technologies and changing user expectations.
By adhering to this systematic approach in our project, we can design a client-side interface that not only meets user requirements but also provides an aesthetically pleasing and responsive user experience using modern web technologies.

6. **Group 2, step-by-step outline on server-side development, including a summary of the purpose for each step chosen.**

Step 1: Project Scope and Requirements Assessment
Establish the project's goals and gather comprehensive requirements. This initial step forms the foundation for the entire development process, ensuring that the server-side functionality aligns with business objectives.

Step 2: Selection of the Technology Stack
Choose suitable server-side technologies, including programming languages (e.g., Python, Node.js, Ruby), frameworks (e.g., Django, Express.js, Ruby on Rails), and databases (e.g., MySQL, MongoDB). This choice significantly impacts development efficiency and performance.

Step 3: Database Design and Schema Creation
Design the database structure and create the schema. This step ensures that data storage and retrieval align with the application's needs and scalability requirements.

Step 4: Server Setup and Configuration
Configure the server environment, covering web servers (e.g., Apache, Nginx) and application servers (e.g., Gunicorn, uWSGI). Configuration includes security settings and server optimization.

Step 5: Routing and API Design
Define the server's routing and API endpoints. Effective API design ensures transparent and consistent communication between the server and client applications.

Step 6: Implementation of Business Logic
Develop code to handle the core business logic of the application, which may include user authentication, data processing, and application-specific functionality.

Step 7: Data Access Layer
Develop the data access layer responsible for interacting with the database. This ensures efficient CRUD (Create, Read, Update, Delete) operations and data validation.

Step 8: Integration of Middleware
Incorporate middleware components to enhance server functionality, covering areas like authentication, security, and request/response handling. Middleware contributes to both functionality and security.

Step 9: Error Handling and Logging
Implement robust error-handling mechanisms to gracefully manage unexpected errors and exceptions. Additionally, establish a logging system to monitor server activity and troubleshoot issues effectively.

Step 10: Security Measures
Enhance server security by incorporating measures like input validation, encryption, and safeguards against common web vulnerabilities (e.g., SQL injection, XSS, CSRF).

Step 11: API Documentation
Generate comprehensive documentation for the server's APIs. This documentation is vital for assisting client developers in understanding how to interact with the server.

Step 12: Testing and Quality Assurance
Conduct thorough testing of the server for functionality, performance, and security. This step addresses any identified bugs or issues during testing.

Step 13: Performance Optimization
Fine-tune the server's performance by optimizing queries, implementing caching mechanisms, and scaling resources as necessary.

Step 14: Deployment
Deploy the server to a production environment, configuring the requisite infrastructure and ensuring scalability, redundancy, and load balancing.

Step 15: Monitoring and Maintenance
Set up monitoring tools to track server performance and address issues in real time. Plan for continuous maintenance and updates to maintain server security and currency.

Step 16: Documentation and Knowledge Transfer
Document the server-side code, configurations, and deployment procedures. This documentation facilitates team collaboration and future maintenance.

By adhering to this systematic approach in our project, server-side development can be conducted efficiently and effectively, ensuring the creation of a resilient and secure backend for web applications.

References

19 music app features to enhance user experience. (2023, July 12). Devabit. https://devabit.com/blog/music-app-features/

Bila, D. (n.d.). A Step-by-Step guide to designing a web application. UGEM. https://ugem.design/blog/web-app-design-process

Server-side website programming first steps - Learn web development | MDN. (2023, July 3). https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 7

Applicable VLOs or EESs for This Week's Case Study

3. Deploy software applications for devices using multiple operating systems.

EES 1.2 Respond to written, spoken, or visual messages in a manner that ensures effective communication. (T, A,)
EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 4.6 Locate, select, organize, and document information using appropriate technology and information systems. (T, A,)
EES 4.7 Analyze, evaluate, and apply relevant information from a variety of sources. (T, A,)

This Week's Detailed Case Study Information

"Well, you all become quite the addition to our team here now, haven't you!" Pitch says as he arrives in the intern workspace bright and early Monday morning, "Over the last few weeks, you have joined your expertise with that of our team's professionals, and it is fair to say you not only impressed me but all of the team as well! As a result, this week, I would like to offer your team your first independent project!"

As the words "independent project" fall out of Pitch's mouth, you do everything you can to remain professional and not just up from your chair with excitement. You have certainly enjoyed learning with Pitch and the rest of the Pockets of Playlists team within the previous week; however, the opportunity to uphold a solo project seems to attest to Pitch's trust in your team's abilities.

"Last week, our team, with your help, orchestrated the creation of a seamless web application ecosystem for our audio streaming services. We knew the time had come to extend our symphony to new heights! Here is the thing though," Pitch's tone shifting, "I think we can take last week's project one step further, and frankly, it is a shame our team, including myself, didn't consider this earlier. Nonetheless, I now see the potential to weave a unified experience across the various platforms. In other words, to craft a Progressive Web App (PWA) version of Pockets of Playlists."

Pitch's idea intrigues you, and you eagerly listen as he explains his thought process further. He shares that the team's meticulous attention to database optimization, client-server interactions, and seamless UI laid the groundwork for what was to come. With this foundation laid, the company can now embrace the power of modern web technologies to "transcend boundaries and deliver a consistent musical experience."

"Again, I truly want to highlight the significance of a PAW in extending the reach of Pocket of Playlists' offerings to users across devices and browsers. And now, before I open this discussion to the group, my final statement is a recommendation to your group. Use the lessons learned in our team's previous endeavour. Build upon our existing web application's architecture, and adapt it to PAW principles. As our earlier work aligned with user requirements, this endeavour will resonate with users' expectations for offline access, fast loading times, and push notifications – hallmarks of a successful PWA."

Deliverables for This Week's Case Study

Your task this week includes:

- Research the key concepts of Progressive Web Apps, including offline access, responsive design, and push notifications.
- Design a tik tok or educational video about the information your team located in the previous task.
- Develop a presentation about PWA tools and technologies and how they may be used with Pockets of Playlists.
- Write a newsletter to the senior stakeholders of the company about how your team will test the PWA's behaviour offline by simulating network disconnection and ensuring your PWA works well on various screen sizes and orientations.

WIL PROJECT
(CASE STUDY)

CSAT-3 (WEEK 7)

CSAT-3 (WEEK-7)
Team Leader: Amrutha Jayanthimala Sukumaran

TEAM MEMBERS
Rahul Ravindra Madeshiya (C0860488)
Nagaraju Tallapelly (C0859913)
Amrutha Jayanthimala Sukumaran (C0860921)
Arvind Singh Jugtwan (C0860886)

1. CONCEPTS OF PROGRESSIVE WEB APPS, INCLUDING OFFLINE ACCESS, RESPONSIVE DESIGN, AND PUSH NOTIFICATION.

INTRODUCTION

Progressive online Apps (PWAs) are a part of online applications that use modern web technologies to give users an experience closer to that of an app. They feature several fundamental features, such as offline access, responsive design, and push notifications, and are made to function on any platform having a browser that complies with standards. By adding these features to our music app "**Pockets of Playlist**," we may improve user experience significantly.

OFFLINE ACCESS

A music app's offline access feature is essential since it enables users to listen to their music without an internet connection. To add offline functionality to our PWA:

Service Workers: These are background-running JavaScript files with network request interception capabilities. They allow us to cache necessary resources such as music files, HTML, CSS, and JavaScript. Our PWA can provide these cached assets to users when they are offline, so previously downloaded music can continue to play.

IndexedDB: Playlists and user preferences can be locally stored on the user's device via IndexedDB, a low-level API for storing massive volumes of structured data. In this manner, people with no internet connectivity can still view their favourite playlists.

Caching Strategies: To regulate how various kinds of content are cached and provided when offline, we can put caching strategies into place. For instance, we can cache playlist data for offline access and broadcast music in real-time using a "Network First, Cache Fallback" approach.

RESPONSIVE DESIGN

One of the key components of PWAs is responsive design. It guarantees that the website will work and look great across a variety of platforms and screen sizes, including PCs, smartphones, and tablets.

Media Queries: To modify the layout and styling of our app according to the user's device, including the screen size and orientation, utilize CSS media queries.

Flexible Grids: Use flexible grid technologies, like as CSS Grid or Flexbox, to design layouts that fluidly adjust to various screen sizes, guaranteeing that the content of the app is laid out and simple to browse.

Mobile-First Approach: If we are creating an app, we have to think about creating it for mobile users first, then gradually improving it for larger screens. By doing this, we can be confident that our software will still work and look good on tiny mobile displays.

PUSH NOTIFICATION

Push notifications can engage users and keep them informed about music updates, playlists, or events in "Pocket of Playlist":

Push API: Even when customers aren't actively using your app, you may still send them alerts to their devices by using the Push API. Users can be informed about future performances by their favourite artists, new music releases, or recommendations for personalized playlists, for instance.

Service Worker Integration: Service staff must receive push alerts. Even when the app is closed, they can still listen for incoming alerts and show them to the user.

Permission Management: Seek consent from users before sending push notifications to respect their privacy. Make sure the user finds the notifications useful and relevant to keep them from unsubscribing.

2. **EDUCATIONAL VIDEO ABOUT THE PREVIOUS TASK DONE BY OUR TEAM.**

Video File Link:
https://mylambton-my.sharepoint.com/:v:/g/personal/c0860488_mylambton_ca/ET4q-qRhxUdIogbUdAj5ZDQBQ37B5BV9MvHF7_BDAvqMng?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZ

3. **POWERPOINT PRESENTATION ABOUT PWA TOOLS AND TECHNOLOGIES, AND HOW THEY WILL BE USED IN POCKETS OF PLAYLISTS.**

PPT File Link
https://mylambton-my.sharepoint.com/:p:/g/personal/c0860488_mylambton_ca/EZ9ekOyofbFNnx4tjwM_HqYByuDVdaCsDS0xRFJVUblnWg?e=TIqk8o

NEWSLETTER CONTINUE NEXT PAGE …

4. NEWSLETTER FOR THE SENIOR STAKEHOLDER.

OCTOBER 27, 2023 *VOLUME 1*

NEWSLETTER

TESTING PWA'S BEHAVIOUR OFFLINE
BY SIMULATING NETWORK DISCONNECTION

MESSAGE TO STAKEHOLDERS

Our team is diligently working to ensure that our PWA functions work seamlessly, even under challenging network conditions and on a wide range of devices and screen orientations.

ONE OF THE KEY ASPECTS OF OUR TESTING STRATEGY INVOLVES EVALUATING HOW OUR PWA BEHAVES WHEN USERS ARE IN AN OFFLINE OR LOW-NETWORK ENVIRONMENT.

BEHAVIOUR TESTING & DESIGN TESTING

Offline Behaviour Testing
Graceful Degradation
Data Synchronization
Error Handling

Responsive Design Testing
Screen Size Compatibility
Orientation Testing
Cross-Browser Compatibility

Thank you for your continued support and trust in our team's capabilities. We will work towards delivering a robust PWA that meets our users' expectations and our company's objectives.

THANK YOU

POCKETS OF PLAYLIST

SharePoint Link: https://mylambton-my.sharepoint.com/personal/c0860488

CONCLUSION

By including key concepts of progressive web apps in "Pocket of Playlist," we hope to develop a dynamic, user-focused interface that responds to the requirements and tastes of users. With our music app, these ideas will improve user satisfaction overall, retention, and engagement and by concentrating on testing processes, we will be able to provide a PWA that not only provides an outstanding online user experience but also satisfies our users' various needs and preferences. Our "Pockets of Playlist" development approach is built around this dedication to quality and user happiness. We're excited to present an exceptional PWA that will please our users.

REFERENCES

Nyakundi, H. (2023, June 27). What is a PWA?
https://www.freecodecamp.org/news/what-are-progressive-web-apps/

A Beginner's Guide To Progressive Web Apps — Smashing Magazine. (2016, August 11).
https://www.smashingmagazine.com/2016/08/a-beginners-guide-to-progressive

Nyakundi, H. (2023, June 27). What is a PWA? Progressive Web Apps for Beginners.
https://www.freecodecamp.org/news/what-are-progressive-web-apps/

Screen orientation. (2023, August 9).
https://www.w3.org/TR/screen-orientation/

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 8

Applicable VLOs or EESs for This Week's Case Study

4. Evaluate and integrate security features into the client and database application tiers to secure against system threats.

EES 2.3 Execute mathematical operations accurately. (T, A,)
EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 4.7 Analyze, evaluate, and apply relevant information from a variety of sources. (T, A,)
EES 5.8 Show respect for diverse opinions, values, belief systems, and contributions of others. (T, A,)

This Week's Detailed Case Study Information

Over the weekend, a team member at the company noticed that the content displayed on Pockets of Playlists' mobile app appeared to be altered. Immediately, this individual recognized that this could indicate that injected scripts are modifying the app's interface. As a result, you and your team enter an uneasy office Monday morning, given the company's potential vulnerabilities, precisely the looming shadow of cross-site scripting (XSS) attacks.

Pitch pairs your team with the company's diligent security team to detect indicators that point to the possibility of malicious actors exploiting this vulnerability. He expresses that this occupancy could potentially compromise the sanctity of user data and undermine the delightful experience that Pockets of Playlists promote. During this, you cannot help but hear the worry in his voice. Rapidly, you and your peers become keenly aware of the gravity of this situation.

"If successful, these scripts can be executed in the context of other users' browsers, leading to unauthorized access, data theft, or other harmful actions," Tony, the lead of the company's security team, shares, "The concern arises from the fact that even a minor security loophole could result in dire consequences for our company and its devoted users.

Your team will now help Tony and the rest of the security team to evaluate the app's client-side codebase to identify any potential XSS vulnerabilities; implement input validation

and output encoding mechanisms to prevent malicious code injection and XSS attacks; and ensure that user-generated content is sanitized before being displayed, ensuring a secure user experience.

Deliverables for This Week's Case Study

Your task this week includes:

- Write a newsletter to the stakeholders of Pockets of Playlists to communicate the company's concerns, including indicators that were "detected" that point to the possibility of malicious actors exploiting this vulnerability
- Develop an educational video about identifying potential XSS vulnerabilities and how this process would appear with a mobile app such as Pockets of Playlists.
- Develop a simple web page that accepts user input, validates it, and displays the sanitized output using input validation and output encoding techniques.
- Create an online portal with access to information about content sanitization techniques.

WIL PROJECT
(CASE STUDY)

CSAT-3 (WEEK 8)

CSAT-3 (WEEK-8)
Team Leader: Rahul Ravindra Madeshiya

TEAM MEMBERS
Rahul Ravindra Madeshiya (C0860488)
Nagaraju Tallapelly (C0859913)
Amrutha Jayanthimala Sukumaran (C0860921)
Arvind Singh Jugtwan (C0860886)

INTRODUCTION

The significance of cybersecurity and online safety in a time when technology affects every aspect of our lives cannot be understated. We are excited to present a comprehensive project that includes a newsletter, a web page, and an instructional video to fulfill this demand and provide our community with important insights on internet security.

- **NEWSLETTER**

  Our newsletter is our source for the latest updates and insights on web security, including the ever-evolving landscape of potential vulnerabilities. We will deliver informative articles, best practices, and real-world examples, ensuring that you stay informed and vigilant in today's digital environment.

- **WEBPAGE SCREENSHOTS**

  We've created a dedicated web page to enhance your experience and provide easy access to essential resources. The web page includes a home page, signup page and login page.

- **EDUCATIONAL VIDEO**

  Awareness is the first line of defence against potential security threats. Our educational video focuses on XSS (Cross-Site Scripting) vulnerabilities, a common and critical threat in web applications. Learn about the dangers of XSS and discover effective strategies to prevent and mitigate these risks.

- **PORTAL FOR CONTENT SANITIZATION TECHNIQUES**

  Ensuring that the content we come upon is secure and free of harmful components is important. Our portal's purpose is to give you useful tips, tricks, and resources for content sanitization. In a time when digital security is critical, our project seeks to provide you with the information and resources needed to safeguard your online identity. Keep checking back for updates regularly, read our newsletter, browse our website, and view our informative video on XSS vulnerabilities to expand your knowledge of web security.

1. NEWSLETTER FOR STAKEHOLDERS

POCKETS OF PLAYLISTS

NEWSLETTER

03 NOV 2023 | MESSAGE TO STAKEHOLDERS ABOUT COMPANY'S CONCERNS

TOP VISIBLE VULNERABILITIES

- DATA LEAKS
- CODE INJECTION
- INSECURE AUTHENTICATION
- SERVER-SIDE VULNERABILITIES
- LACK OF ENCRYPTION
- LACK OF SECURITY UPDATES
- THIRD-PARTY LIBRARIES
- USER PRIVACY

DATA SECURITY FOR POCKETS OF PLAYLISTS

IN TODAY'S DIGITAL AGE, SECURING OUR DATA TAKES ON CRITICAL SIGNIFICANCE. ADHERING TO RECOMMENDED SECURITY PROTOCOLS BECOMES THE CORNERSTONE OF SAFEGUARDING OUR PERSONAL INFORMATION AND CONFIDENTIAL DATA. THIS INCLUDES THE CREATION OF STRONG AND UNIQUE PASSWORDS AND ENABLING TWO-FACTOR AUTHENTICATION FOR OUR VARIOUS ACCOUNTS. CONSISTENTLY UPDATING SOFTWARE AND APPLICATIONS, ALONG WITH THE USE OF FIREWALLS AND ANTIVIRUS SOLUTIONS, ACTS AS A SHIELD AGAINST VULNERABILITIES AND MALICIOUS SOFTWARE. ENCRYPTION SERVES AS AN ADDITIONAL LAYER OF PROTECTION, WHILE REGULAR DATA BACKUPS ARE VITAL FOR SWIFT RECOVERY IN UNFORESEEN SITUATIONS. EXERCISING VIGILANCE WHEN DEALING WITH EMAILS AND HYPERLINKS, STRENGTHENING WI-FI NETWORKS, AND ROUTINELY REVIEWING APP PERMISSIONS ARE ALL INTEGRAL COMPONENTS OF DATA SECURITY. BY ADOPTING THESE MEASURES AND STAYING INFORMED ABOUT EMERGING THREATS, WE CAN SIGNIFICANTLY ENHANCE THE SECURITY OF OUR DATA.

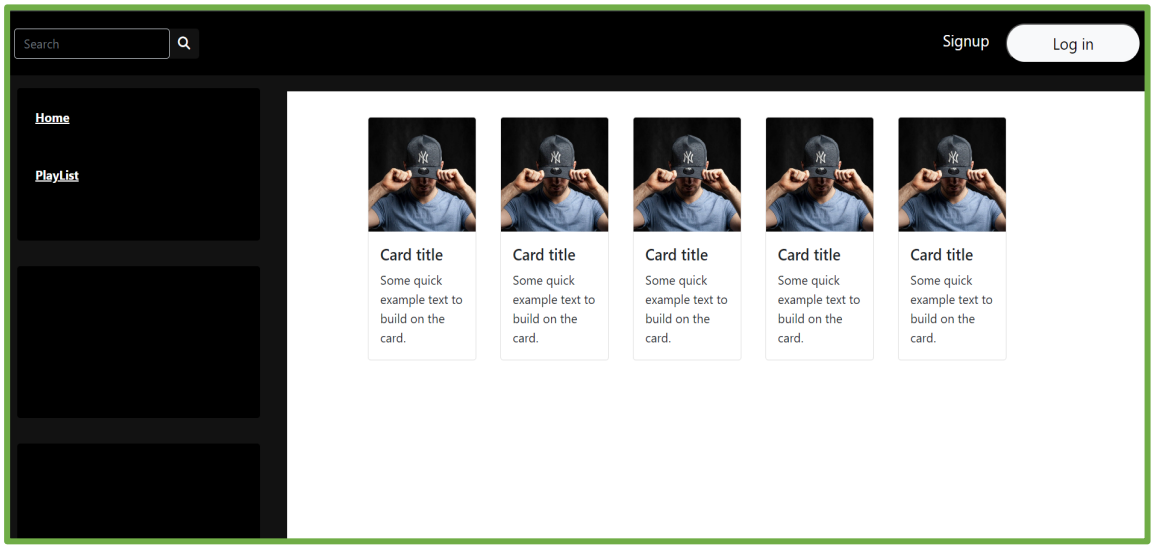
THANK YOU

2. EDUCATIONAL VIDEO

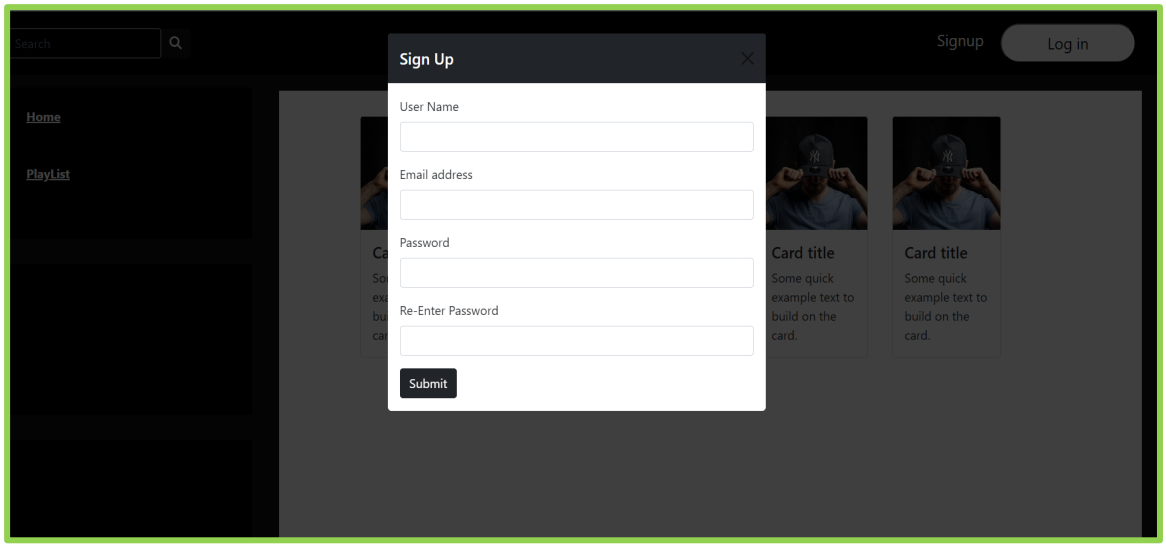
Video Link – https://mylambton-my.sharepoint.com/

3. WEBPAGE SCREENSHOTS FOR OUR POCKETS OF PLAYLIST WEBSITE

Screenshot 01: This is the homepage for our "POCKETS OF PLAYLIST" website

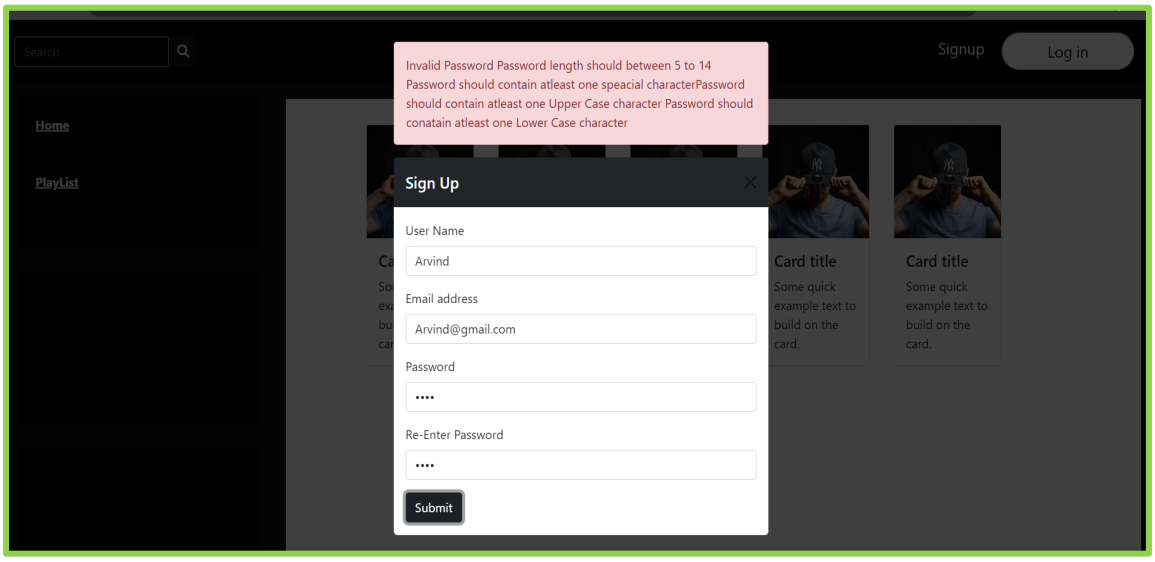


Screenshot 02: This is the signup page, we can fill in the data to create our accounts.

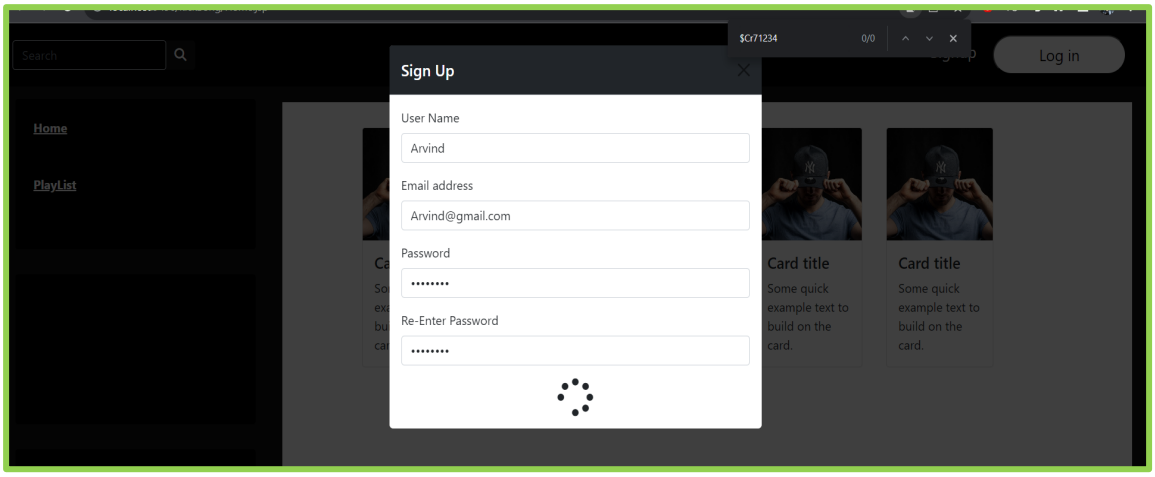


SCREENSHOT CONTINUES - NEXT PAGE

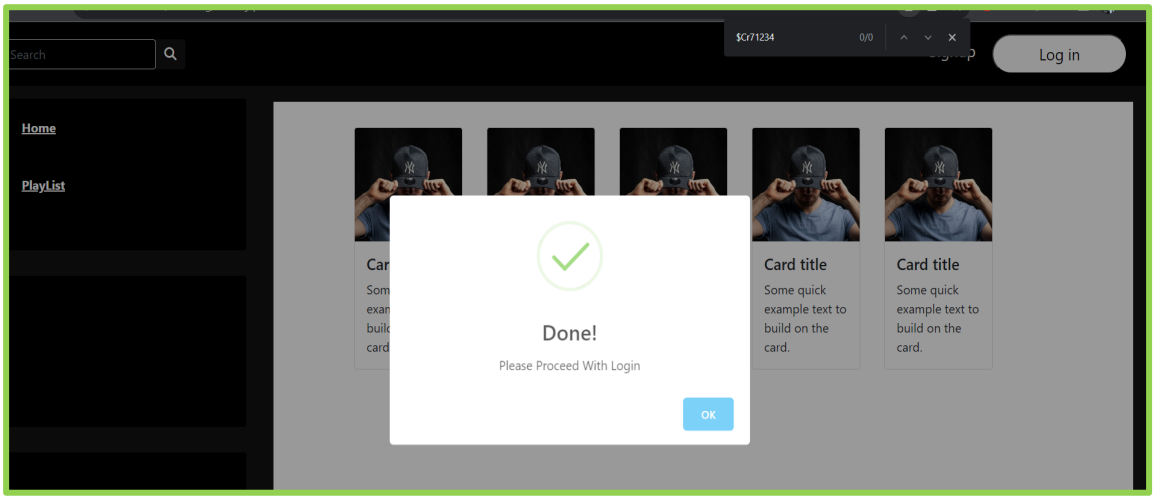
Screenshot 03: There are some validations displayed when we add different passwords. This helps to verify the correct user login.



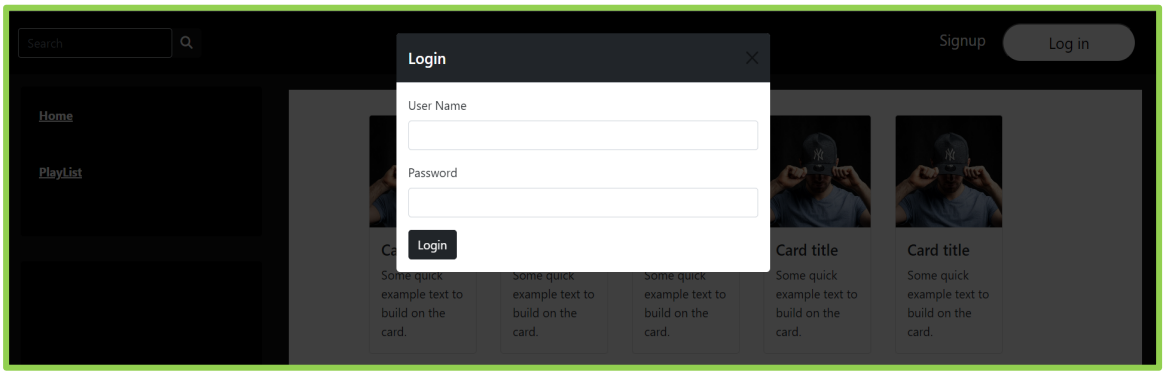
Screenshot 04: When the filled data is accurate, a user account will created.



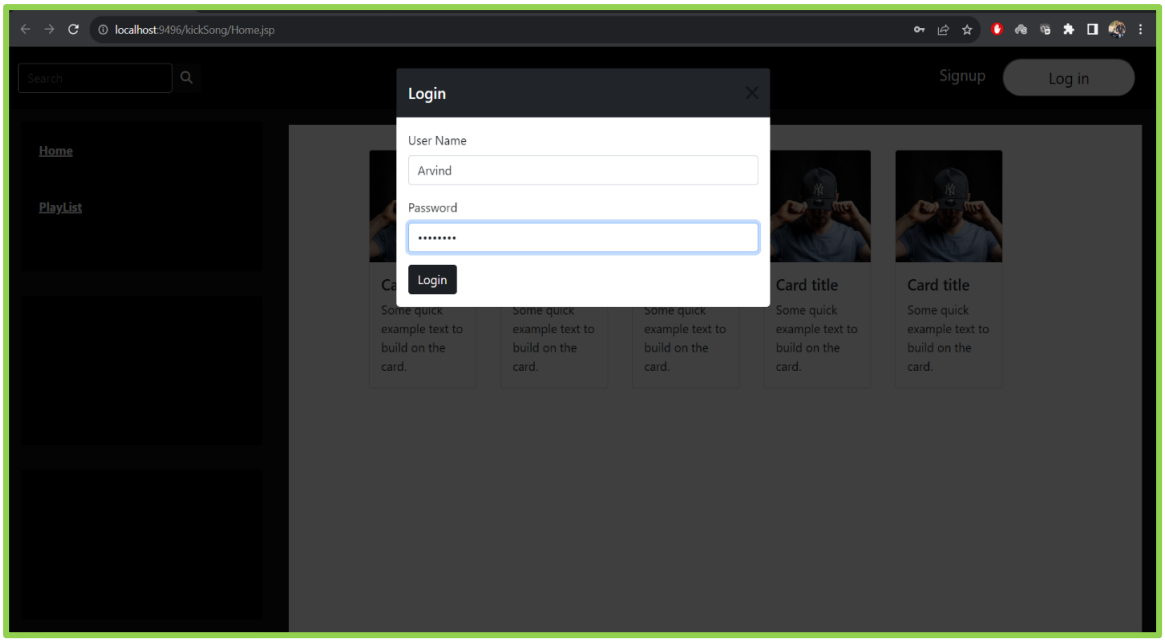
Screenshot 05: After the account is created, the user can see the pop-up message displaying "Done!"



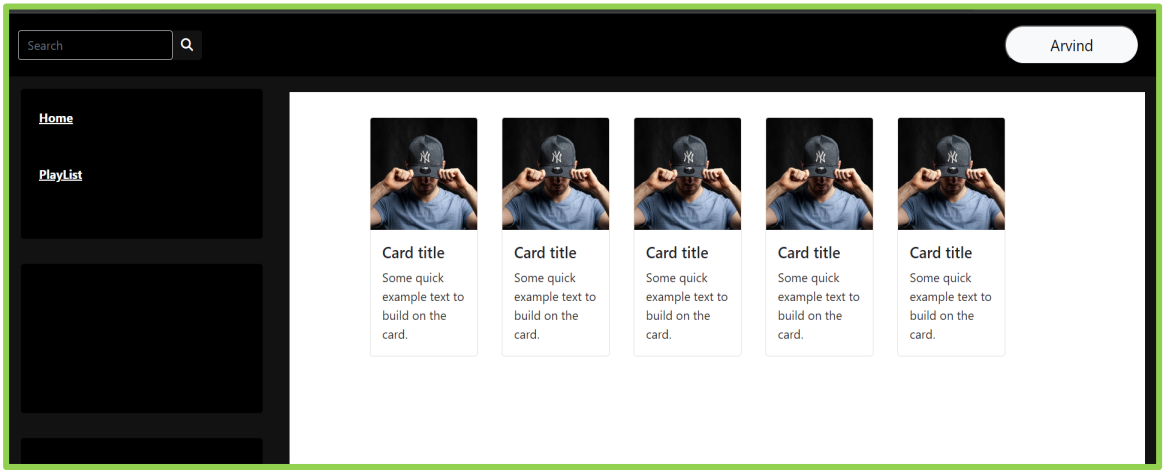
Screenshot 06: This is the login page. Users need to add a username and password and click on the Login button.



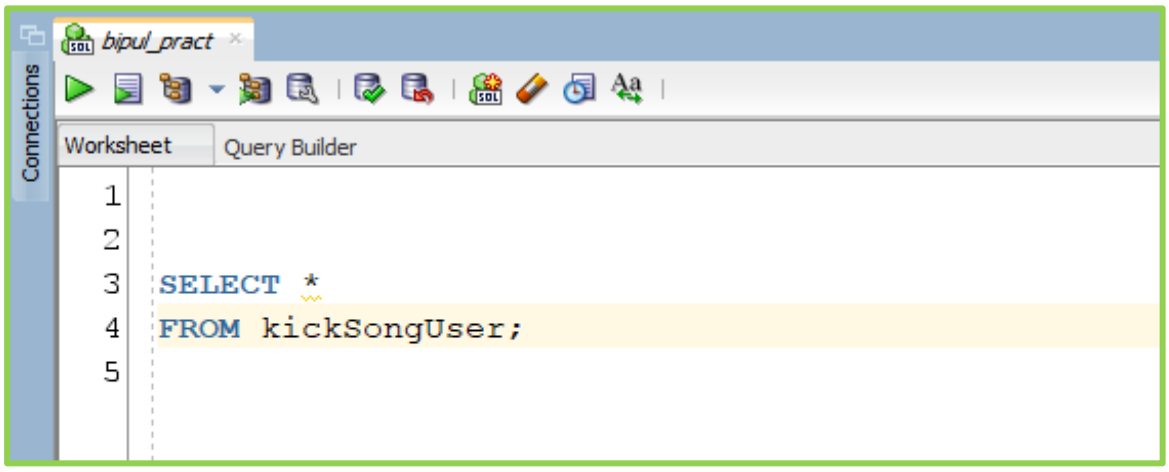
Screenshot 07: After filling in the data, the user needs to click on the login button.



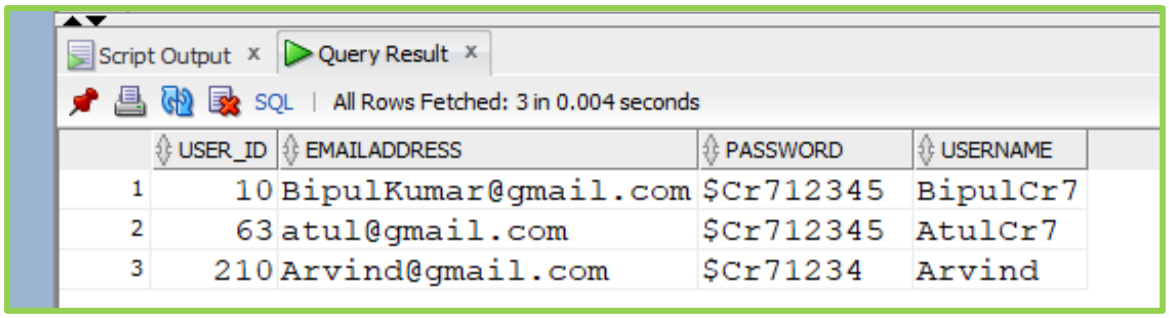
Screenshot 08: This is the home page, after successful login user can see their name on the top right corner of the webpage.



Screenshot 09: SQL query for displaying data from the database.



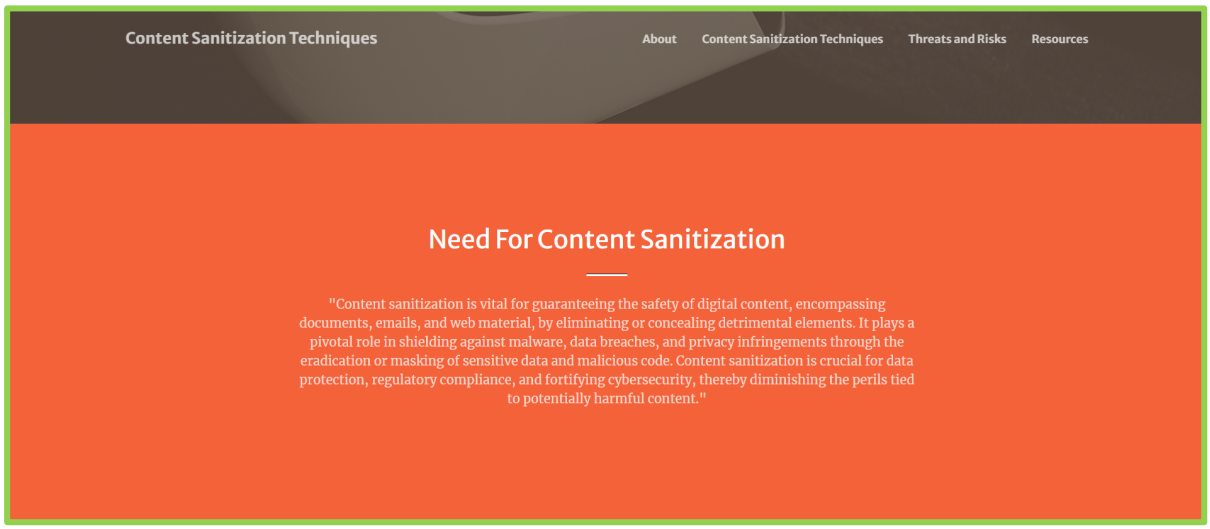
Screenshot 09: Displaying user data in the database.



4. PORTAL FOR CONTENT SANITIZATION TECHNIQUES (SCREENSHOTS)

Screenshot 01: Content Sanitization Introduction.

Screenshot 02: Why Content Sanitization is needed.



Screenshot 03: Content Sanitization Techniques.

Screenshot 04: Content Sanitization Threats and Risks.

Screenshot 05: Content Sanitization Threats and Risks Continued.

6. **Compatibility Issues**

Some content sanitization procedures can alter files in a manner that causes compatibility problems with specific software or platforms, resulting in user frustration and operational disruptions.

7. **User Error**

Human errors during the sanitization process, such as misconfiguring settings, can lead to incomplete or ineffective content sanitization.

8. **Resource Intensiveness**

Certain content sanitization methods can consume significant system resources, affecting overall system performance and efficiency.

9. **False Positives**

Excessive content sanitization may incorrectly label legitimate content as malicious or sensitive, leading to the removal of vital data or information.

Content Sanitization Risks

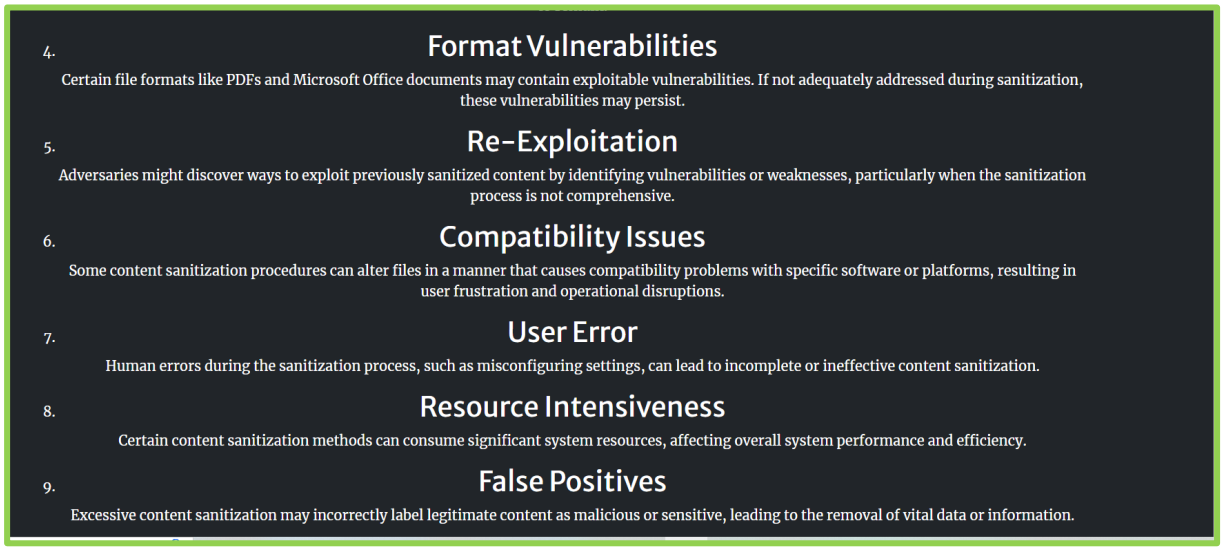
1. **Incomplete Sanitization**

Insufficiently removing all malicious or sensitive elements from files or documents during the sanitization process, potentially leaving residual data that poses a security risk.

2. **Data Loss**

Excessive content sanitization methods can lead to the deletion of legitimate data, rendering files unusable. This can disrupt business operations and create data recovery challenges.

Screenshot 06: Content Sanitization Threats and Risks Continued.



4. **Format Vulnerabilities**

Certain file formats like PDFs and Microsoft Office documents may contain exploitable vulnerabilities. If not adequately addressed during sanitization, these vulnerabilities may persist.

5. **Re-Exploitation**

Adversaries might discover ways to exploit previously sanitized content by identifying vulnerabilities or weaknesses, particularly when the sanitization process is not comprehensive.

6. **Compatibility Issues**

Some content sanitization procedures can alter files in a manner that causes compatibility problems with specific software or platforms, resulting in user frustration and operational disruptions.

7. **User Error**

Human errors during the sanitization process, such as misconfiguring settings, can lead to incomplete or ineffective content sanitization.

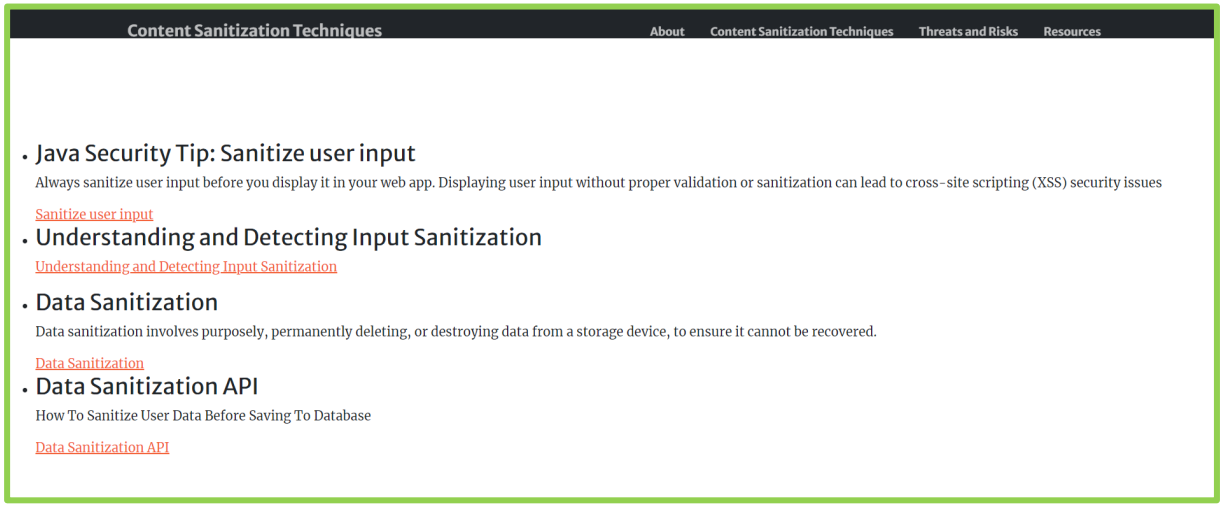
8. **Resource Intensiveness**

Certain content sanitization methods can consume significant system resources, affecting overall system performance and efficiency.

9. **False Positives**

Excessive content sanitization may incorrectly label legitimate content as malicious or sensitive, leading to the removal of vital data or information.

Screenshot 07: Content Sanitization Threats and Risks Continued.



Content Sanitization Techniques

About | Content Sanitization Techniques | Threats and Risks | Resources

- **Java Security Tip: Sanitize user input**

Always sanitize user input before you display it in your web app. Displaying user input without proper validation or sanitization can lead to cross-site scripting (XSS) security issues

Sanitize user input

- **Understanding and Detecting Input Sanitization**

Understanding and Detecting Input Sanitization

- **Data Sanitization**

Data sanitization involves purposely, permanently deleting, or destroying data from a storage device, to ensure it cannot be recovered.

Data Sanitization

- **Data Sanitization API**

How To Sanitize User Data Before Saving To Database

Data Sanitization API

CONCLUSION

To sum up, we designed a newspaper to present the ideas we emphasized and discussed in our newsletters regarding the company's concerns. We then identified the possible method to indicate the malicious actors who exploit this vulnerability. Further, we created an education video explaining the detailed version of XSS, which is cross-site scripting, and its significance. In technical aspects, we worked on creating a simple webpage that accepts user input, validates it, displays the sanitized output, and finally, we created an online portal to showcase the content of sanitization techniques.

REFERENCES

Idsc-Admin. (2023, November 3). Data Sanitization Terminology and Definitions. International Data Sanitization Consortium. https://www.datasanitization.org/data-sanitization-terminology/

Kalman, G. (2014, May 29). 10 common web security vulnerabilities. Toptal Engineering. https://www.toptal.com/cyber-security/10-most-common-web-security-vulnerabilities

Welekwe, A., & Welekwe, A. (2022, July 19). How to find XSS vulnerability. Comparitech. https://www.comparitech.com/net-admin/how-to-find-xss-vulnerability/

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 9

Applicable VLOs or EESs for This Week's Case Study

1. Evaluate system requirements and implement multi-tiered (client, server, and database) web applications to meet client requirements.

EES 1.1 Communicate clearly, concisely, and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience. (T, A,)
EES 1.2 Respond to written, spoken, or visual messages in a manner that ensures effective communication. (T, A,)
EES 2.3 Execute mathematical operations accurately. (T, A,)
EES 5.8 Show respect for diverse opinions, values, belief systems, and contributions of others. (T, A,)

This Week's Detailed Case Study Information

As you've realized in the past week with Pockets of Playlists, the company is always searching for ways to enhance its services to benefit its clients. As a result, it comes as no surprise to you as you discover your team's project for the upcoming week.

You enter the intern workspace and sit at your desk to begin your work day. Scrolling through your computer, you find an email in your inbox forwarded to you by Pitch. You notice that the original email is from Donny Fry, an architect with the company. Donny writes:

"Hi Pitch! I have come across an understanding of our dear users' requirements, and I believe you must also come to know them. Moreover, I have begun to ponder our team's routes to meet these very needs."

Donny's opening line to his email certainly intrigues you. You scan through the rest at a rapid pace, eager to understand. You learn that the challenge is to provide users with an enriched experience of sharing and collaborating on playlist curation. Donny reiterates that the platform's current limitations have ignited a fire to innovate in a world where music is not merely heard but shared.

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Just as you finish reading the conclusion of Danny's email, Pitch pops his head into the room, greeting you and your peers.

"I am hopeful you all have seen the email I sent you?" Your team nods in response, "Perfect. Well, it is without further ado that I assign your team a wonderful case for the week. You have become responsible for helping Pockets of Playlists with our aspiration to enable users to share their playlists with friends and collaborate on playlist curation. As you know, the current platform lacks the seamless social sharing functionality users desire. So please, develop features that allow users to share playlists through social media platforms and generate unique shareable links. *And,*" He dramatically scans the room, "Implement real-time collaborative editing for shared playlists, allowing multiple users to add and modify songs collaboratively."

Deliverables for This Week's Case Study

Your task this week includes:

- In a list, identify the client requirements.
- Write a newsletter to the senior stakeholders of the company to explain the process of social sharing implementation as it relates to the project at hand.
- Design a real-time collaborative playlist editing system, considering user interactions and data synchronization. Choose appropriate technologies (e.g., WebSockets) for real-time communication.
- Collaborate to plan for conflicts in a video when multiple users edit the same playlist concurrently. Consider how your team could gracefully implement conflict resolution mechanisms to handle conflicting changes.

WIL PROJECT
(CASE STUDY)

CSAT-3 (WEEK 9)

CSAT-3 (WEEK-9)
Team Leader: Nagaraju Tallapelly

TEAM MEMBERS
Rahul Ravindra Madeshiya (C0860488)
Nagaraju Tallapelly (C0859913)
Amrutha Jayanthimala Sukumaran (C0860921)
Arvind Singh Jugtwan (C0860886)

INTRODUCTION

In the ever-evolving realm of audio streaming and music services, "Pockets of Playlists" stands out as an innovative platform meticulously crafted with a distinct purpose: to elevate the global music-sharing experience for passionate enthusiasts. With a steadfast dedication to simplicity, ingenuity, and community, our platform transforms the way music aficionados engage, explore, and exchange their love for enduring classics and the most recent tracks. Pockets of Playlists distinguishes itself through a combination of inventive features and active community involvement. Recognizing the profound personal and social nature of music, our platform embodies this awareness by creating an environment where users can not only curate, explore, and share their preferred songs but also transcend the limitations of conventional music streaming.

1. CLIENT REQUIREMENTS

Client requirements for a music editing application may vary depending on their specific needs and vision for the software. However, here are some general features and functionalities that clients might prioritize when seeking a music editing app:

User-Friendly Interface
A design that is easy to use, allowing users to navigate the app effortlessly. Clear and accessible controls for manipulating and editing music tracks.

Audio Editing Tools
Features such as cutting, copying, pasting, and deleting audio clips. Capability to adjust volume levels and apply fade-in/out effects. Support for time-stretching and pitch-shifting.

Multi-track Editing
The capacity to work with multiple audio tracks simultaneously. Simple management of layers and arrangement of various tracks.

Real-time Collaboration
Functionality that facilitates collaboration among multiple users in real-time. Synchronization of edits across all collaborators.

Effects and Filters
A variety of audio effects, including reverb, equalization, and modulation. Filters designed to enhance or modify the sound, such as high-pass and low-pass filters.

Instrumentation and Virtual Instruments
Integration of virtual instruments or the option to import MIDI data. Support for a diverse range of instrument sounds and presents.

File Compatibility
Support for various audio file formats such as MP3, WAV, and AIFF. Import and export capabilities to seamlessly integrate with other software.

Version Control
Mechanisms for tracking changes and versions of a project. Ability to revert to previous versions or save multiple iterations of a project.

Cloud Integration
Options for storage and backup through cloud services. Accessibility to projects from multiple devices.

Security Measures
Robust authentication and authorization to protect intellectual property. Encryption of data to ensure privacy and security.

Platform Compatibility
Compatibility with various operating systems, including iOS, Android, Windows, and macOS. Responsive design catering to different screen sizes.

User Permissions
Detailed control over user access and permissions within collaborative environments. Assignment of different roles for project contributors (e.g., editor, viewer).

Export and Sharing Options
Capability to export projects in various formats. Integration with social media platforms or other sharing options.

Feedback and Support
In-app support or a help centre for users to find guidance. Mechanisms for users to provide feedback or report issues.

Comprehending these requirements and engaging in close collaboration with the client will facilitate the development of a music editing app that aligns with their expectations and meets the needs of the intended user base.

CLIENT REQUIREMENTS FOR PLATFORM ENHANCEMENT

Seamless Social Sharing
Enable users to share playlists effortlessly on popular social media platforms. Implement functionality for the generation of unique shareable links.

Real-time Collaborative Editing
Introduce a feature for real-time collaborative editing of shared playlists. Allow multiple users to simultaneously add and modify songs on collaborative playlists.

These improvements are anticipated to meet user preferences and cultivate a heightened level of interactivity and social connectivity in the music-sharing experience on our platform.

2. NEWSLETTER

POCKETS OF PLAYLISTS
NEWSLETTER

10 NOVEMBER 2023

SOCIAL SHARING IMPLEMENTATIONS

Sharing music on social media apps can be a great way to express your musical preferences, discover new songs, and connect with others who share similar tastes. The process of sharing music on social apps may vary depending on the platform.

SHARING MUSIC

- Login into the application
- Select the music
- Select the Share Option
- Choose the Platform
- Privacy Settings
- Post or Share

MESSAGE TO STAKEHOLDERS

In conclusion, sharing music playlists adds a dynamic and personalized dimension to the way we connect with others through music. Whether it's introducing friends to favourite tracks, setting the mood for an event, or collaborating on a playlist, the process is made seamless through various music streaming platforms.

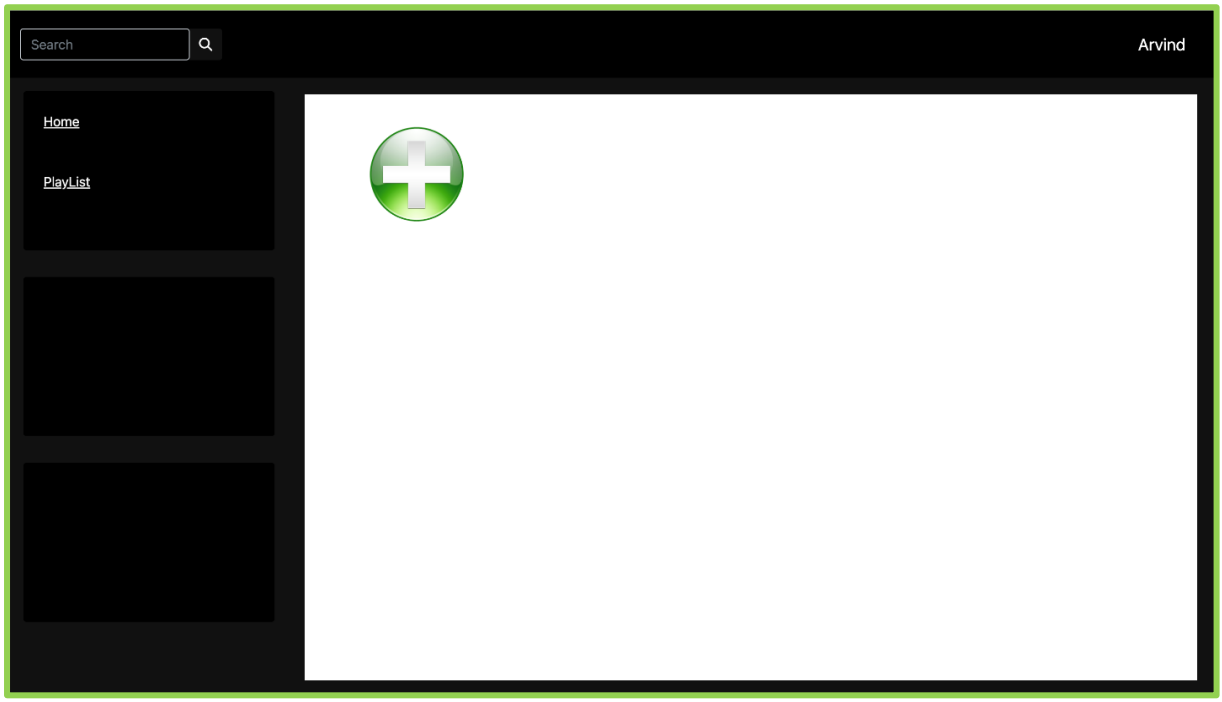
KEY POINTS IN SHARING MUSIC

- Ensure that the accounts are properly linked for a seamless sharing experience.
- Respect copyright and licensing rules when sharing music.
- Explore additional features like collaborative playlists or sharing entire playlists.

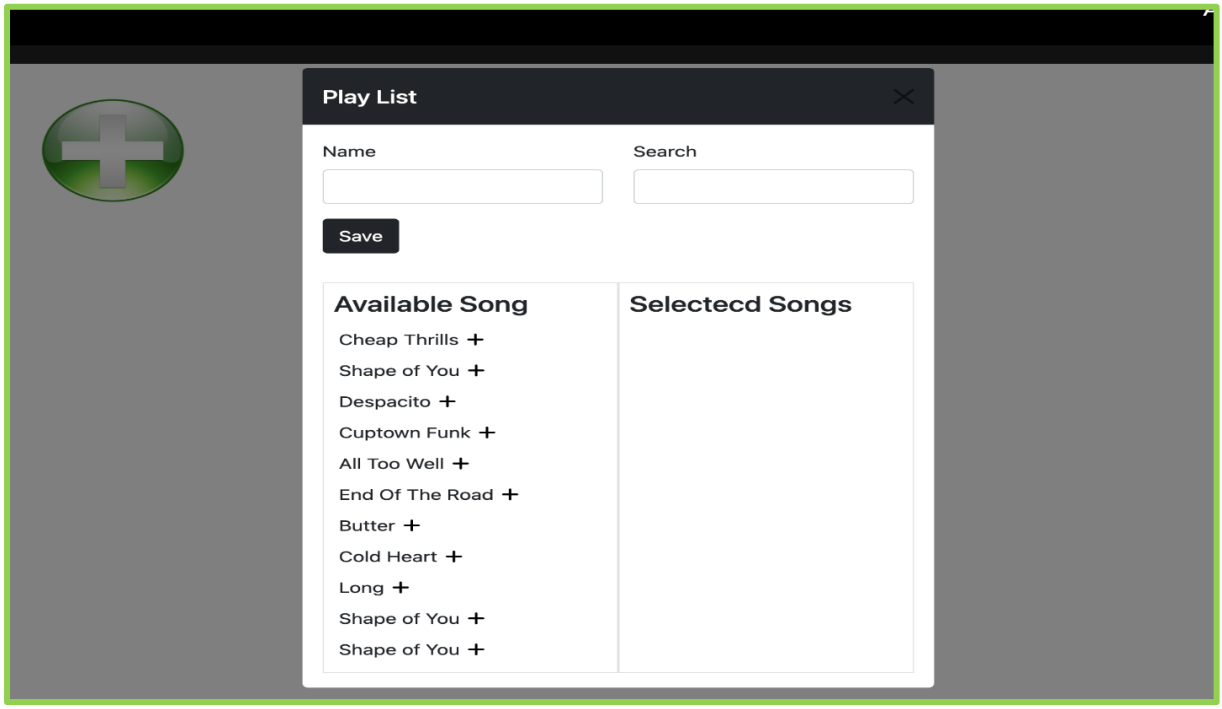
THANK YOU

3. REAL-TIME COLLABORATIVE PLAYLIST EDITING SYSTEM (SCREENSHOTS)

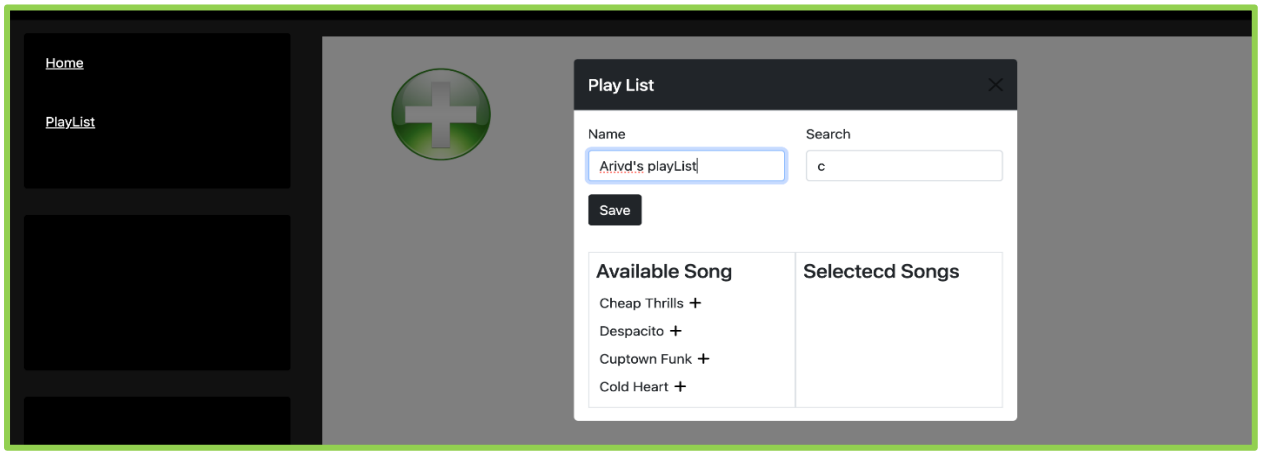
SCREENSHOT 01: Access the App using Arvind's credentials.



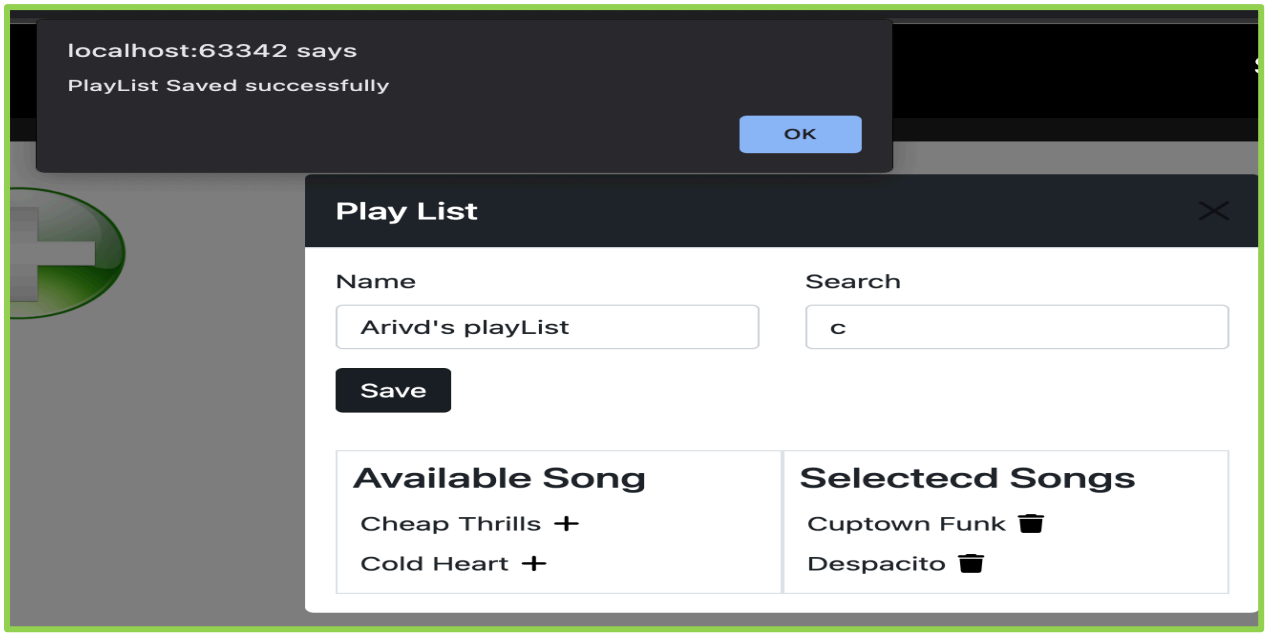
SCREENSHOT 02: Upon clicking the "add" button, users are presented with the default playlist. This display comprises two sections: "Available Songs" and "Selected Songs." Users can choose their preferred songs from the available song section, and the selected songs will be showcased in the designated area.



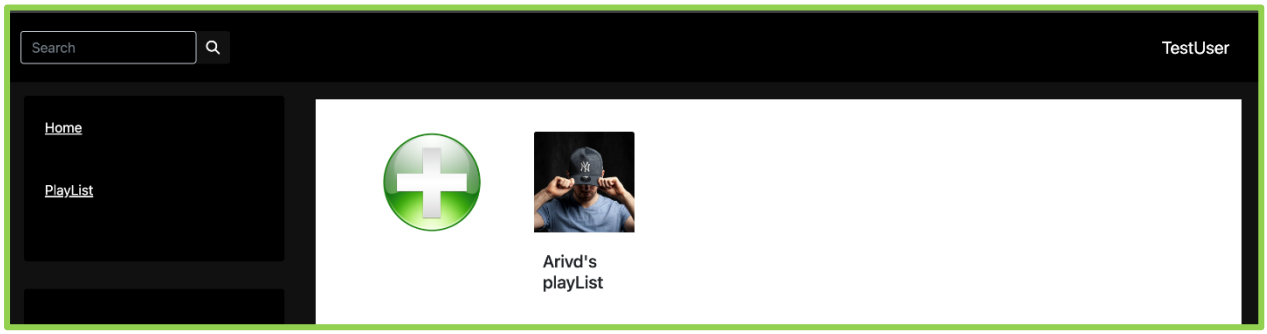
SCREENSHOT 03: In the interface, there are two search boxes. The "Name" text box allows us to create a name for our playlist, while the "Search" text box enables us to search for songs.



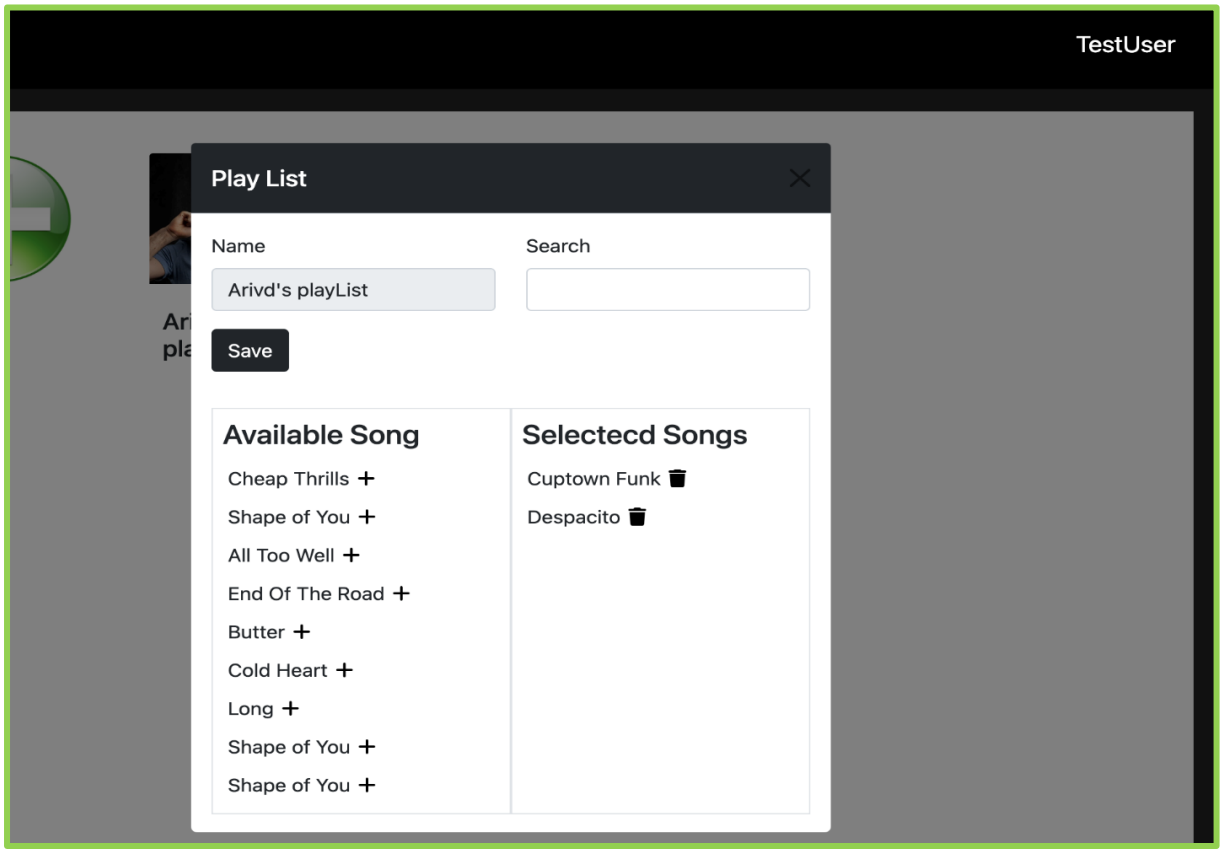
SCREENSHOT 04: Once the user has created their favourite song playlist, they can proceed to click the 'Save' button to store the playlist.



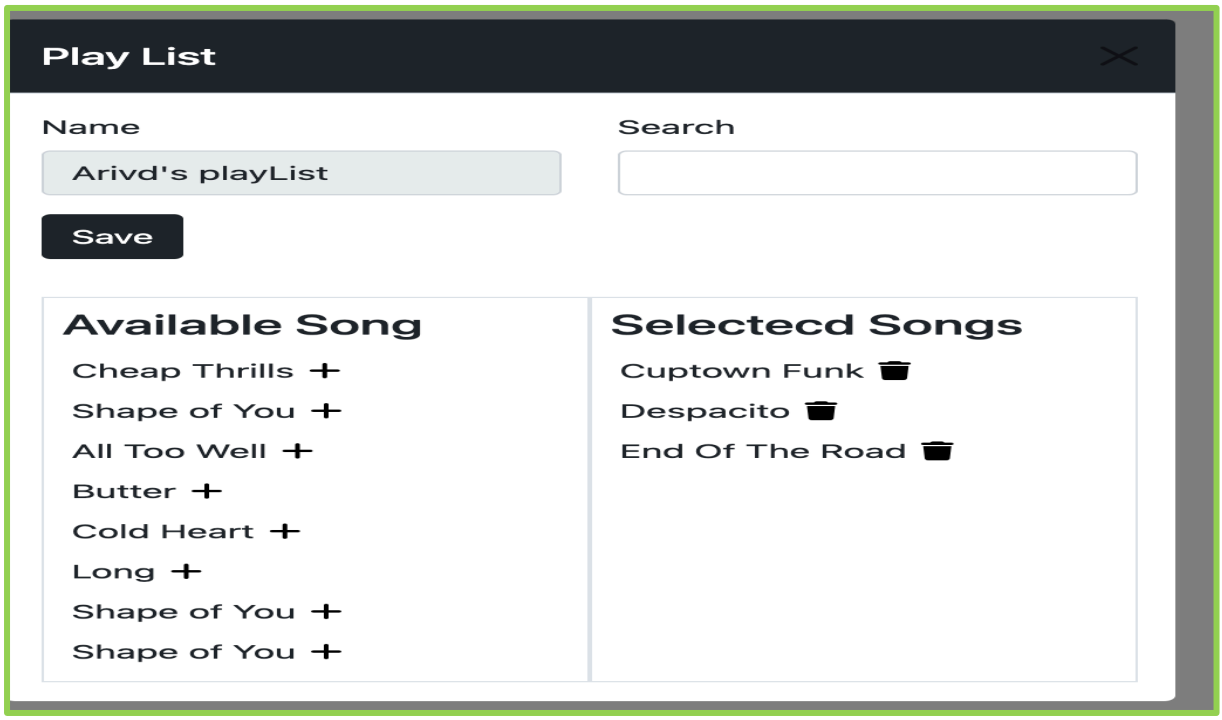
SCREENSHOT 05: If another user, such as "**TestUser**," logs in, the playlist previously created, namely '**Arvind's Playlist**,' will be displayed.



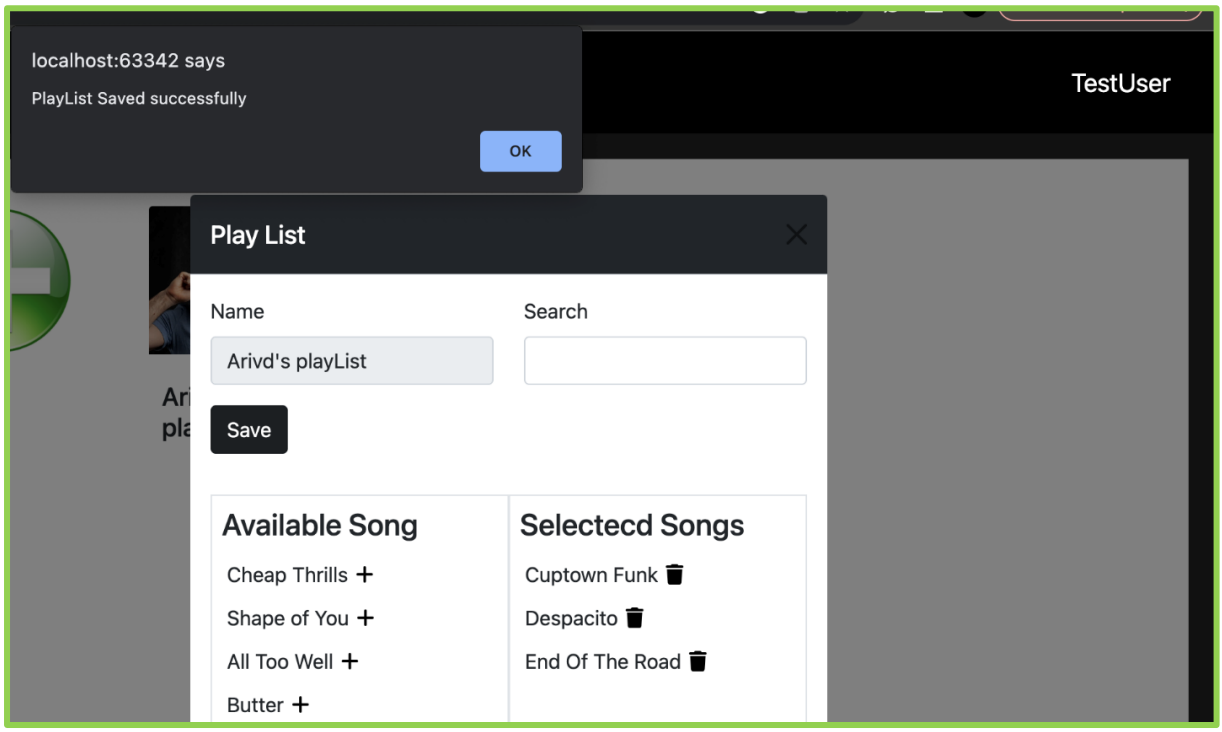
SCREENSHOT 06: Next, "**TestUser**" will select "**Arvind Playlist**" and his favourite songs of Arvind will be displayed in the "Selected Songs" section



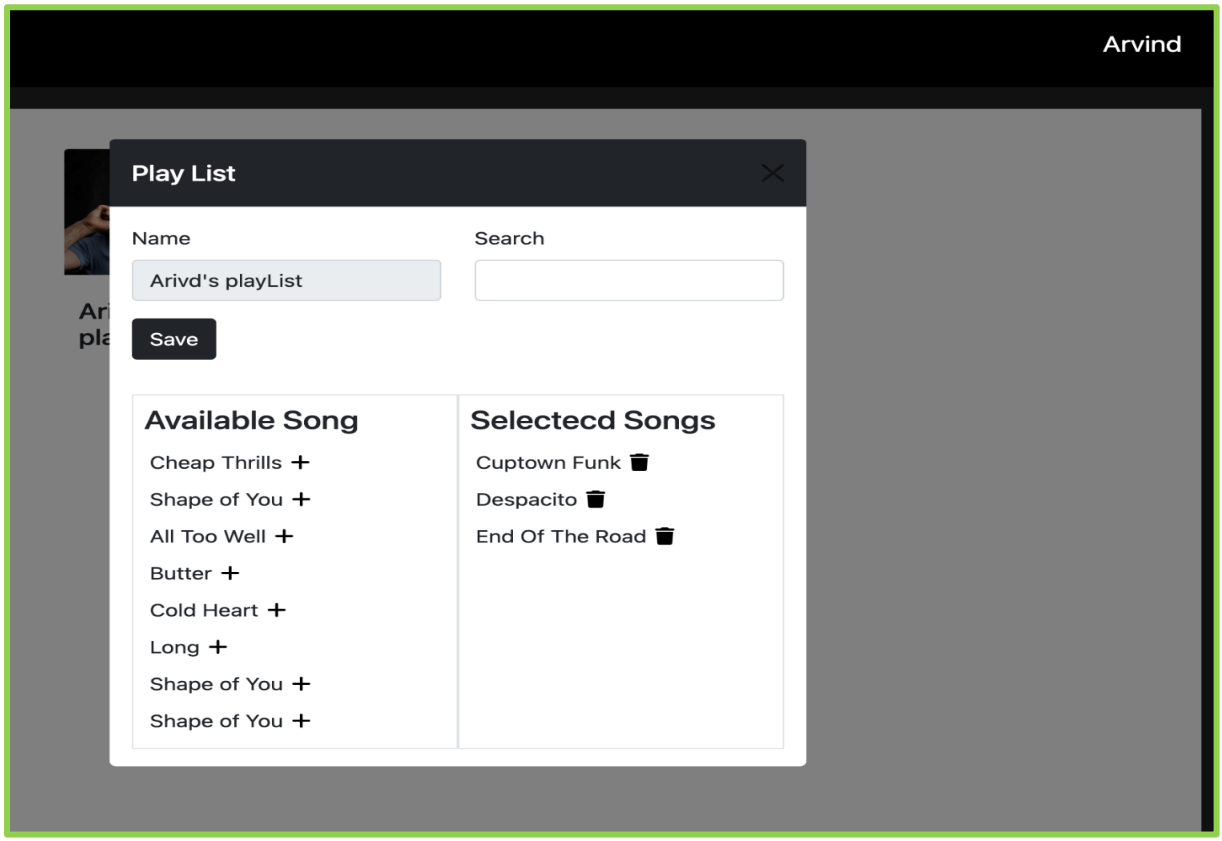
SCREENSHOT 07: TestUser can make modifications to Arvind's Playlist, such as adding a song to the "Selected Songs" section



SCREENSHOT 08: Save the modifications by selecting the "Save" button



SCREENSHOT 09: Any alterations made by "**TestUser**" will be reflected in Arvind's Playlist when Arvind logs in and reviews the playlist



4. TEAM VIDEO DISCUSSING CONFLICTS WHEN MULTIPLE USERS EDIT THE PLAYLIST CONCURRENTLY

Team Video Link: https://mylambton-my.sharepoint.com/:v:/g/personal/c0860488

CONCLUSION

In conclusion, the development of a real-time collaborative music player aligned with the client's specifications represents a compelling opportunity to transform how users interact with and craft music. Through the integration of features such as concurrent editing, live collaboration, and seamless communication tools, the envisioned music player seeks to establish itself as a vibrant and interactive platform catering to both musicians and enthusiasts.

The client's focus on intuitive interfaces, robust security protocols, and compatibility with various file formats underscores a dedication to delivering a flexible and user-friendly product. Furthermore, the incorporation of version control and detailed user permissions is geared towards facilitating a fluid and effective collaborative experience, fostering creativity and cooperation among users in real time.

As we embark on the developmental journey, it is imperative to remain attuned to emerging technologies and user input. Regular updates and feature enhancements will be vital to ensuring the music player's continued relevance and competitiveness within the ever-changing digital landscape.

In summary, the proposed real-time collaborative music player has the potential to redefine the landscape of music creation and collaboration. It aspires to be a robust and user-friendly platform that addresses the diverse needs of musicians and artists. By meeting the specified client requirements, this project seeks to play a role in advancing the field of music production, nurturing a community of creators who can collaborate seamlessly and bring their musical visions to fruition.

REFERENCES

CANVA: Visual suite for everyone. (n.d.-a). https://www.canva.com/

Client requirements. Client requirements - Designing Buildings. (n.d.). https://www.designingbuildings.co.uk/wiki/Client_requirements

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: CSAM

Week 10

Applicable VLOs or EESs for This Week's Case Study

2. Design, model, implement, maintain, and query databases using an enterprise-level relational database management system (DBMS) to satisfy end-user specifications.

EES 1.1 Communicate clearly, concisely, and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience. (T, A,)
EES 1.2 Respond to written, spoken, or visual messages in a manner that ensures effective communication. (T, A,)
EES 3.5 Use a variety of thinking skills to anticipate and solve problems. (T, A,)
EES 5.9 Interact with others in groups or teams in ways that contribute to effective working relationships and achieving goals. (T, A,)

This Week's Detailed Case Study Information

You and your peers are fired up as you enter the office today. Last week, you were extremely successful in your endeavours with the company's mobile app. This success was met with praise from many employees of Pockets of Playlists – which certainly felt fulfilling.

As you practically skip to your desk, you notice a document laid upon it. Without hesitation, you snatch the document from its place and read its title, "Weaving the Melodic Tapestry - Redesigning Pockets of Playlists' Database." Based on the choice of language used in the title, you immediately assume that this document must have been given to you by your lovely *and certainly unique* supervisor, Pitch.

You continue reading to learn that the company aims to revolutionize their playlist creation by repeatedly accommodating the intricate demands of Pockets of Playlists' users. This time, these enthusiasts yearn for playlists that transcend genres, "interweaving moods and artist preferences into harmonious tapestries of sound." Sadly, however, while reliable, the company's current database schema lacks the flexibility to orchestrate such complex musical compositions.

WIL PROJECT

Program Name: Computer Software & Database Development

Project Code: **CSAM**

He makes your team's mission clear, to design, model, implement, maintain, and query a database that satisfies the end user's hunger for unique playlist creation experiences that blend genres, moods, and artist preferences. To do so, your team must analyze Pockets of Playlists' user stories and design a flexible database schema that accommodates dynamic playlist structures. Additionally, you must utilize data types to store diverse playlist attributes, enabling intricate song compositions.

Deliverables for This Week's Case Study

Your task this week includes:

- Create a list of user stories that capture the unique playlist creation features desired by users.
- Analyze each user story to understand the specific attributes, genres, moods, and artist preferences that must be accommodated in the database.
- Design a new database schema that supports dynamic and intricate playlist structures. While considering their relationships, they define tables for playlists, songs, artists, genres, and other relevant entities.
- Research and identify appropriate data types for storing diverse playlist attributes.
- Develop a presentation to present your process of each previous task to your supervisor. This may be done in any format you wish.

WIL PROJECT
(CASE STUDY)

CSAT-3 (WEEK 10)

CSAT-3 (WEEK-10)
Team Leader: Arvind Singh Jugtwan

TEAM MEMBERS
Rahul Ravindra Madeshiya (C0860488)
Nagaraju Tallapelly (C0859913)
Amrutha Jayanthimala Sukumaran (C0860921)
Arvind Singh Jugtwan (C0860886)

INTRODUCTION

Creating a music streaming platform that caters to a wide range of customer interests is a difficult but rewarding endeavour. Our objective is to develop and integrate unique playlist features that are highly meaningful to our users. Understanding user stories, optimizing database architecture, selecting acceptable data formats, and successfully presenting our results are all part of this endeavour. The choice of proper data types is critical in guaranteeing effective storage and manipulation of various playlist characteristics. A thorough investigation is required to identify the most appropriate data formats that are linked with the different properties of playlist elements. Our unwavering goal is to build and execute a playlist creation system that not only meets but surpasses user expectations, giving all users with a personalized and immersive musical experience.

1. LIST OF USER'S STORIES THAT CAPTURE UNIQUE PLAYLIST FEATURES

STORY 01: Users want to be able to collaborate with friends on playlists, allowing numerous participants to add, delete, and alter the music in real time.

STORY 02: Music fans want an AI-powered recommendation system that can create dynamic playlists based on their listening history, mood, and time of day.

STORY 03: Fitness fans want a tool that would automatically produce playlists customized to their training regimens, altering pace and genre based on activity intensity.

STORY 04: Commuters want to make playlists that change based on their GPS locations so that the music matches their environment or travel path.

STORY 05: Users with various musical tastes want the ability to easily blend numerous playlists into one cohesive list with no duplicate tracks.

STORY 06: Party hosts want a feature that allows guests to make live song requests, which are seamlessly integrated into the playlist queue.

STORY 07: Language learners want to be able to access curated playlists with songs in the language they're learning, which steadily increase in complexity as they progress.

STORY 08: Individuals seeking nostalgia may choose to create playlists based on certain life eras or milestones, gathering songs from specific years or key life events.

STORY 09: Curators require a tool that can assess the mood or energy of existing playlists and offer complementary music to improve the overall vibe.

STORY 10: Users who want to save data choose an offline option that allows them to pre-download or cache playlists for offline listening.

2. ANALYSING EACH USER'S STORIES TO UNDERSTAND SPECIFIC ATTRIBUTES, GENRES, MOODS, AND ARTIST PREFERENCES

STORY 01: Collaborative Playlists
Attributes: Real-time collaboration to add, delete, and alter music.
Genres: Support for a different variety of music genres.
Moods: Flexibility for accommodating various moods based on users.

STORY 02: AI-powered Recommendation System
Attributes: AI-driven dynamic playlist creation based on listening history, mood, and time of day.
Genres: Analyzing the user's all-time listening history.
Moods: Adaptation to the user's current mood.
Artists: Recognition of most preferred artists based on users' listening history.

STORY 03: Fitness Customized Playlists
Attributes: Automatic playlist generation based on workouts, analyzing their moments.
Genres: Including a wide variety of genres which are suitable for different workout routines.
Moods: Enthusiastic and motivational music for different workouts.

STORY 04: Location-Based Playlists for Commuters
Attributes: Playlists change based on locations.
Genres: Adaptation to the user's surroundings or travel zones like specific songs for mountains or temples.
Moods: Customizing based on the user's location and activity.

STORY 05: Blending Multiple Playlists
Attributes: Capability to mix various playlists into one connected list.
Genres: Compatibility of different genres.
Moods: Ensuring seamless transitions between different moods.

STORY 06: Live Song Requests at Parties
Attributes: Feature that can accommodate guests to make live song requests.
Genres: adaptability to provide various guest preferences.
Moods: Energetic atmosphere for parties**.**

STORY 07: Language Learning Playlists
Attributes: Playlists carefully curated with songs in the language you're studying.
Genres: A diverse selection to accommodate different musical preferences within the chosen language.
Moods: Including various levels of complexity to provide the steps for language learning.

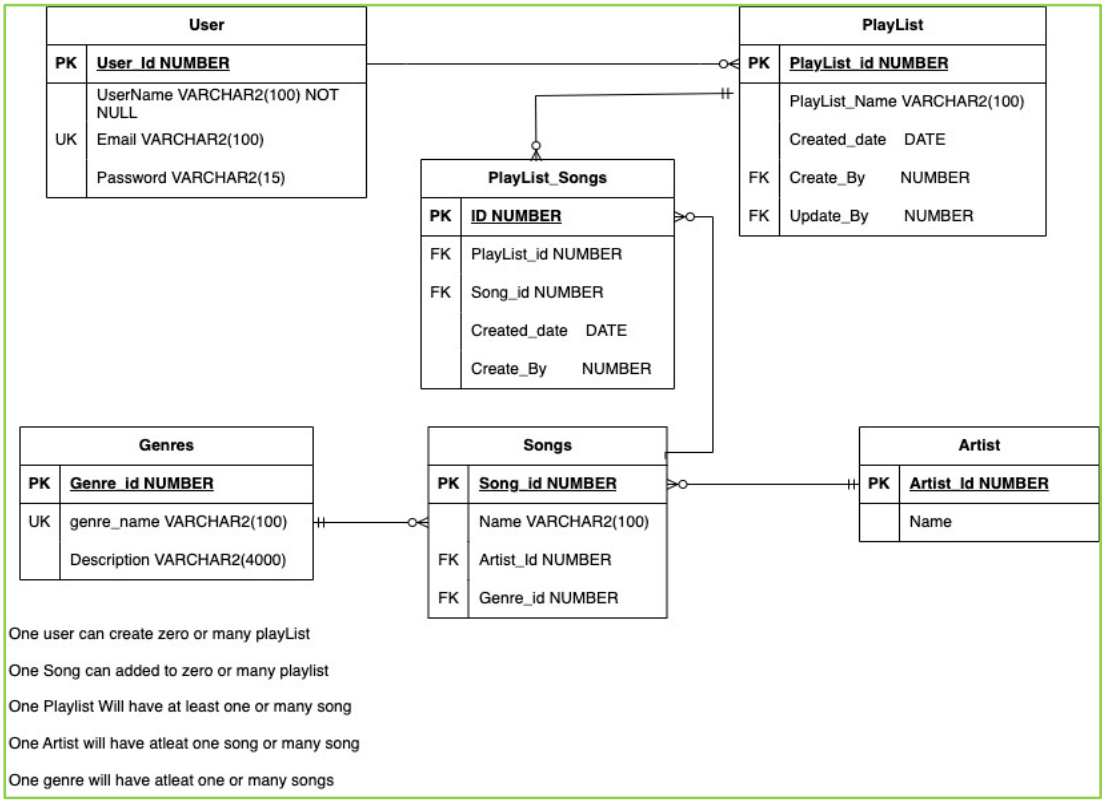
STORY 08: Nostalgia-Based Playlists
Attributes: Development organized according to life stages or significant milestones
Genres: Match your favourite music styles from different life stages.
Moods: Feel nostalgic, capturing the vibe of important life moments.

STORY 09: Mood Assessment and Complementary Music
Attributes: Tool to see how playlists make you feel.
Genres: Adding similar music styles to make the vibe better.
Moods: Changing to lift the mood of the playlist.

STORY 10: Saving Data with Offline Mode
Attributes: Downloading playlists for listening without using data
Genres: Picking music you like for the saved playlists.
Moods: Reflective of the moods represented in saved users' playlists

3. DATABASE SCHEMA AND TABLES FOR PLAYLISTS, SONGS, ARTISTS, GENRES, AND OTHER RELEVANT ENTITIES

DATABASE ENTITY-RELATIONSHIP DIAGRAM



DATABASE TABLES

TABLE 01: User
Attributes: User_id (NUMBER), UserName (VARCHAR2), Email(VARCHAR2), Password (VARCHAR2).

TABLE 02: PlayList
Attributes: PlayList_id (NUMBER), PlayList_Name (VARCHAR2), Created_date (DATE), Created_By (NUMBER), Updated_By (NUMBER).

TABLE 03: PlayList_Songs
Attributes: ID (NUMBER), PlayList_id (NUMBER), Songs_id (NUMBER), Created_date (DATE), Created_By (NUMBER)

TABLE 04: Songs
Attributes: Song_id (NUMBER), Name (VARCHAR2), Artist_id (NUMBER), Genre_id (NUMBER).

TABLE 05: Genres
Attributes: Genre_id (NUMBER), genre_name (VARCHAR2), Description (VARCHAR2).

TABLE 06: Artist
Attributes: Artist_id (NUMBER), Name (VARCHAR2).

4. DATATYPE FOR SHARING DIVERSE PLAYLIST ATTRIBUTES

User Table

COLUMN	DATATYPE
User Id	Number
Username	Varchar
Email	Varchar
Password	Varchar

PlayList Table

COLUMN	DATATYPE
PlayList_id	Number
PlayList_Name	Varchar
Created_date	Date
Create_By	Number
Update_By	Number

PlayList_Songs Table

COLUMN	DATATYPE
ID	Number
PlayList_id	Number
Song_id	Number
Created_date	Date
Created_By	Number

Genre Table

COLUMN	DATATYPE
Genre_id	Number
Genre_name	Varchar
Description	Varchar

Songs Table

COLUMN	DATATYPE
Song_id	Number
Name	Varchar
Artist_id	Number
Genre_id	Number

Artist Table

COLUMN	DATATYPE
Artist_id	Number
Name	Varchar

5. POWERPOINT PRESENTATION FOR PROGRESS OF EACH PREVIOUS TASK

PPT File Link: https://mylambton-my.sharepoint.com/:p:/g/personal/c0860886

CONCLUSION

In reviewing our efforts to improve playlist-building tools, it is clear that knowing user stories is critical in creating an immersive musical experience. Our careful investigation of each user narrative revealed certain qualities such as genres, moods, artist preferences, and more, all of which are essential for an enhanced musical trip. Choosing appropriate data types was critical to ensuring the efficient storage and handling of various playlist attributes. Our research precisely identified appropriate data types that corresponded to the various characteristics of playlist elements. Our commitment to creating an engaging music platform remains unwavering, fueled by our desire to evolve and innovate in response to user needs, fostering seamless integration between technology and music enthusiasts.

REFERENCES

Gareth. (2023, June 27). 45 User Story Examples To Inspire Your Agile Team. Parabol. https://www.parabol.co/blog/user-story-examples/

draw.io - free flowchart maker and diagrams online. (n.d.). https://app.diagrams.net/

Kurata, D. (2023, March 7). Variable data types explained. freeCodeCamp.org. https://www.freecodecamp.org/news/variable-data-types-explained/

Microsoft. (n.d.). Free Trial - Try Microsoft 365 for a month - Microsoft Store. https://www.microsoft.com/en-ca/microsoft-365/try?culture=en-ca&country=ca