VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA-590014

Industry Internship Report

on

“FULL STACK WEB DEVELOPMENT”

Submitted by

Ms. ARYA A	**4DM21CS011**
Ms. ISHA I V	**4DM21CS020**
Ms. JASNITHA JOY	**4DM21CS021**
Ms. MINAL FATHIMA.P	**4DM21CS026**

UNDER THE GUIDANCE OF

Mr. Guruprasad G
Asst Professor,
Dept.of CSE

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

YENEPOYA
INSTITUTE OF TECHNOLOGY

YENEPOYA INSTITUTE OF TECHNOLOGY

N.H.13, THODAR, MOODBIDRI-574225, MANGALORE, D.K.

2023-24

YENEPOYA INSTITUTE OF TECHNOLOGY

THODAR, MIJAR POST, MOODBIDRI-574225

(Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Industry Internship report entitled **"FULL STACK WEB DEVELOPMENT"** is an authentic record of the work carried out by **Ms.ARYA A,** 4DM21CS011**, Ms.ISHA I V,** 4DM21CS020**, Ms.JASNITHA JOY ,**4DM21CS021**, Ms.MINAL FATHIMA P ,** 4DM21CS026, students of 5th semester in partial fulfillment of requirements for the award of Bachelor's Degree in **Computer Science & Engineering** prescribed by Visvesvaraya Technological University during the year **2023-24**

____

Signature of the Guide

(Mr.Guruprasad G)

____

Signature of the HOD

(Dr Manjunath Kamath)

External Viva

Name of the examiner

1.__________________

2. __________________

Signature with Date

1. __________________

2.__________________

ACKNOWLEDGEMENT

The successful completion of any work would be incomplete without a mention of the people who made it possible, whose constant guidance and encouragement served as a beacon light and crowned our efforts with success. We owe our gratitude to many people who helped and supported us during our Project "**Music Player Website**".

Our deepest thanks to our guide **Mr. Guruprasad G**, Asst. Professor, Dept. of CSE, Yenepoya Institute of Technology for his constant support , encouragement and providing us with the necessary advices and help. We are highly indebted to him/her for taking keen interest in our work, monitoring and providing guidance throughout the completion of our work.

We also thank **Dr Athokpam Bikramjit Singh.,** Professor and Industry Internship Coordinator, Department of Computer Science & Engineering for his constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr Manjunath Kamath,** Professor & Head of the Department of Computer Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. R G. D'Souza**, Principal, Yenepoya Institute of Technology for his constant support and providing us with all the facilities that were required.

Finally, yet importantly, we express our heartfelt thanks to our family & friends for their wishes, encouragement and providing me moral strength for the successful seminar presentation.

ARYA A	**4DM21CS011**
ISHA I V	**4DM21CS020**
JASNITHA JOY	**4DM21CS021**
MINAL FATHIMA P	**4DM21CS026**

ABSTRACT

This internship opportunity focuses on Full stack web development. Full stack web development is a multifaceted approach to creating interactive and dynamic websites or web applications. This comprehensive skill set covers both the front-end and back-end aspects of development, enabling individuals to handle the entire software development process. The term "full stack" refers to the complete range of technologies and frameworks a developer should be proficient in, ensuring a seamless integration of various components.

On the front-end, a full stack developer must master HTML for structuring content, CSS for styling, and JavaScript for creating interactive user interfaces. This encompasses the visual elements users interact with, emphasizing responsiveness and a polished user experience. Conversely, the back-end involves server-side programming, database management, and server configuration. Full stack developers commonly work with server-side languages like Node.js, Python, Ruby, or PHP to handle requests and process data. They are adept at working with databases such as MySQL, PostgreSQL, or MongoDB, ensuring efficient storage and retrieval of information.

The full stack developer acts as a bridge between these two realms, seamlessly connecting the user interface with the server-side logic. Application programming interfaces (APIs) play a crucial role in this integration, allowing communication between different software components. Understanding RESTful API design is fundamental for effective data exchange.

Moreover, full stack developers need proficiency in version control systems like Git, ensuring collaboration and code management. They must also be familiar with deployment and hosting platforms, such as Heroku or AWS, to make the developed applications accessible to users.

In essence, full stack web development demands adaptability and continuous learning. The ever-evolving landscape of technologies requires developers to stay updated with the latest frameworks and tools. This abstract underscores the holistic nature of full stack web development, emphasizing the need for a well-rounded skill set to create robust and user-friendly digital experiences.

CONTENTS

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

Inventeron Technologies, spearheaded by CEO Muhammed Misbah, is a tech powerhouse in Bangalore, redefining industry landscapes. With expertise spanning AI, web development, automotive solutions, and IoT, the company is a sought-after destination for businesses craving innovation. Muhammed Misbah's leadership fosters a dynamic, collaborative culture, propelling the company to the forefront of tech excellence. From reshaping automotive tech to pioneering IoT solutions, Inventeron stands as a beacon of innovation. Their commitment extends to nurturing tech talent through hands-on internships, providing real-world experience in AIML, web development, and more. With a culture that encourages innovation and a dedication to exceeding expectations, Inventeron Technologies is architecting the digital future with creativity and expertise.

Full-stack web development is the comprehensive approach of building web applications, encompassing both front-end and back-end development. On the front end, developers create the user interface and experience using languages like HTML, CSS, and JavaScript. This involves designing responsive and interactive elements that users interact with directly. On the back end, developers focus on server-side logic, databases, and server deployment. Common back-end technologies include Node.js, Python and Java.

A full-stack developer is adept at both ends of the web development spectrum, facilitating seamless communication between the user interface and the server. They integrate databases, manage user authentication, and handle server-side operations. Understanding the full stack enables developers to create dynamic and feature-rich web applications. Tools like Git for version control and various frameworks simplify the development process. Collaboration, adaptability, and problem-solving skills are vital for a full-stack developer, as they navigate through the entire web development lifecycle, from conceptualization to deployment. Embracing this holistic approach empowers developers to craft robust, scalable, and user-friendly web applications.

CHAPTER 2 : LITERATURE SURVEY

Web development encompass research on web design principles, usability studies, responsive web design trends, the impact of page load speed on user experience, and the evolution of web development frameworks. It could also include web security, accessibility standards, and the implications of emerging technologies like Progressive Web Apps (PWAs) or Web Assembly. Different programming languages and frameworks contribute to efficient and scalable web applications.Full-stack web development would involve examining research on front-end technologies such as HTML, CSS, and JavaScript, back-end frameworks like Node.js, Django, or Ruby on Rails, and databases such as MySQL or MongoDB.

Bootstrap is a popular front-end framework that simplifies web development by providing pre-designed HTML, CSS, and JavaScript components. It helps create responsive and visually appealing websites with ease. Bootstrap, an open-source front-end framework, revolutionizes web development by providing a comprehensive toolkit for building responsive and visually appealing websites. Originally developed by Twitter, Bootstrap facilitates the creation of consistent and mobile-friendly user interfaces. Its key features include a responsive grid system, predefined CSS styles, and JavaScript plugins. The framework's modular nature allows developers to easily customize and extend its components, saving significant time in the design and coding process. Bootstrap's popularity is driven by its flexibility, making it suitable for projects of various scales. With a vast community and extensive documentation, developers can harness Bootstrap to create modern, cross-browser compatible websites that adapt seamlessly to different screen sizes. Whether you're a novice or an experienced developer, Bootstrap simplifies the complexities of web development and empowers creators to deliver polished, user-friendly experiences.

CHAPTER 3 : PROBLEM STATEMENT AND SOLUTION STRATEGY

3.1 PROBLEM STATEMENT:

Design a full-stack web application to streamline the process of managing tasks within a collaborative team environment. The system should allow users to create, assign, and track tasks, set priorities, and monitor progress. The application must feature user authentication and authorization mechanisms, ensuring secure access to different levels of functionality based on user roles. Additionally, implement a responsive and intuitive user interface for seamless navigation on various devices. The back end should handle data storage, retrieval, and updates, while the front end should dynamically display task information and updates in real-time. Integration of third-party APIs for additional functionalities, such as notifications or file attachments, would enhance the application's utility. The goal is to create a robust and user-friendly full-stack solution that enhances team collaboration and task management efficiency.

3.2 SOLUTION STRATEGY:

To address the outlined problem statement in full stack web development, follow this solution strategy:

1. Requirements Analysis:
   - Conduct a thorough analysis of user requirements and expectations for task management.
   - Identify key features, user roles, and integration points.

2. Technology Stack:
   - Choose a suitable technology stack for both front-end and back-end development.
   - Consider frameworks like React or Angular for the front end and Node.js or Django for the back end.

3. Database Design:
   - Design a scalable and efficient database schema to store task-related data.
   - Choose a database system that aligns with the project's needs (e.g., MySQL, MongoDB)

4. User Authentication and Authorization:

- Implement secure user authentication mechanisms.
- Set up role-based access control to ensure appropriate permissions for different user roles.

5. Front-end Development:

- Develop a responsive and intuitive user interface using the chosen front-end framework.
- Implement real-time updates for task progress using technologies like WebSocket.

6. Back-end Development:

- Build a robust back-end system to handle task creation, assignment, and tracking.
- Implement RESTful APIs for communication between the front end and back end.

7. Integration of Third-party APIs:

- Explore and integrate third-party APIs for additional features, such as notifications or file attachments.

8. Testing:

- Conduct thorough testing, including unit testing, integration testing, and user acceptance testing.
- Ensure the application is secure, scalable, and functions as expected.

9. Deployment:

- Deploy the application on a reliable hosting platform or cloud service.
- Configure continuous integration and deployment pipelines for streamlined updates.

10. Monitoring and Maintenance:

- Implement monitoring tools to track application performance and user engagement.
- Provide regular maintenance and updates based on user feedback and changing requirements.

CHAPTER 4 : PROPOSED SYSTEM

As part of learning to code, we completed a project that involved creating a basic Music player website using HTML, CSS, and JavaScript. This project was designed to showcase our skills in front-end web development. It is a responsive and user-friendly web application that allows users to play music. We used HTML and CSS to design and style the interface, and We implemented JavaScript to handle user input and interaction such as basic playback controls -play, pause, and next. We also gained experience working with version control tools such as Git and GitHub.

A music player website aims to provide users with an immersive and user-friendly platform for listening to and discovering music. The website typically features a sleek and intuitive user interface that allows users to play a vast library of songs. Key features include user customization options. Responsive design ensures a seamless experience across various devices. Users can explore different genres, create personalized playlists, and enjoy a continuous playback experience. Additionally, integration with social media platforms may enhance the sharing and discovery of music. The website combines functionality with an engaging design to create a compelling digital space for music enthusiasts to enjoy and explore diverse musical content.

CHAPTER 5 : SYSTEM REQUIREMENTS ANALYSIS AND SPECIFICATION

5.1 HARDWARE REQUIREMENTS:

- **Windows 11:** Windows 11, released by Microsoft in October 2021, is the latest iteration of the Windows operating system. It introduces a refreshed and centered Start Menu, taskbar, and system tray, emphasizing a more modern and streamlined design. The new Snap layouts and Snap Groups enhance multitasking, allowing users to organize and switch between open windows efficiently.

  Windows 11 supports Android apps through the Microsoft Store, providing a broader application ecosystem. The integration of Microsoft Teams directly into the taskbar facilitates seamless communication and collaboration. Performance improvements, such as better virtual desktop support and gaming enhancements with Direct Storage and Auto HDR, contribute to an overall enhanced user experience.

- **Internet:** The internet is a global network of interconnected computers and devices that communicate through standardized protocols. It enables the exchange of information, data, and resources across the globe. Developed over decades, the internet has become an integral part of modern life, revolutionizing communication, commerce, education, and entertainment.

  Key components of the internet include websites, online services, and communication tools such as email and instant messaging. The World Wide Web (WWW) is a major application on the internet, providing a vast collection of interlinked documents and multimedia content.

  The internet operates on a decentralized model, with no single governing body. Instead, various organizations and protocols ensure its smooth functioning. Internet protocols, such as TCP/IP, form the backbone for data transmission.

5.2 SOFTWARE REQUIREMENTS:

- **VS Code:** Visual Studio Code (VS Code) can be a valuable tool for developing a music player website by providing a feature-rich and efficient development environment. VS Code offers a powerful code editor with features like syntax highlighting, autocompletion, and code snippets. This aids developers in writing clean and error-free code for the frontend (HTML, CSS, JavaScript) and backend. It includes an integrated terminal, allowing developers to run commands directly within the editor, it seamlessly integrates with version control systems like Git, it supports a vast library of extensions, it comes with built-in debugging tools for various languages. Developers can set breakpoints, inspect variables, and debug JavaScript code, which is essential for troubleshooting and optimizing the functionality of a music player website. The Live Server extension in VS Code allows developers to launch a local development server with live reload capabilities.

- **HTML:** HTML (Hyper Text Markup Language) plays a fundamental role in building the structure and content of a music player website. HTML defines the overall structure of the web pages, including the layout, headers, footers, and main content area. It establishes the foundation for organizing various components of the music player website. Semantic HTML elements, such as <header>, <nav>, <main>, <footer>, and others, provide a meaningful structure to the content. HTML5 introduced the <audio> element, which is crucial for embedding audio content, such as music tracks, directly into web pages. It allows specifying the audio file source, controls, and other attributes for a seamless music playback experience. If the music player website includes a playlist feature, HTML is used to structure the list of tracks. This could involve using ordered or unordered lists (<ol>, <ul>) and list items (<li>) to present the playlist in an organized manner. In summary, HTML is the backbone of a music player website, defining the structure, content, and interactive elements that contribute to a seamless user experience.

- **JavaScript:** JavaScript is crucial in a music player website for enhancing interactivity, managing dynamic content, and creating a smooth user experience. JS is used to control the playback of audio elements. This involves functions to play, pause, stop, adjust volume, and skip tracks. JS can be utilized to dynamically load content, such as playlists or additional tracks, without requiring a full page reload. JS is involved in handling user authentication and authorization processes. Local Storage in JavaScript allows the website to store small amounts of data on the user's deviceIn summary, JavaScript enhances the functionality and user experience of a music player website by enabling dynamic content updates, managing audio playback, and facilitating seamless interactions between the user and the application.

- **CSS:** CSS (Cascading Style Sheets) is crucial in a music player website for styling and enhancing the visual presentation of the user interface.CSS is used to define the visual styling of various elements, such as buttons, navigation bars, backgrounds, and fonts. CSS is employed to create a responsive design that adapts to different screen sizes and devices. For the audio player interface, CSS is used to style and customize the appearance of playback controls, progress bars, volume sliders, and other elements to ensure a cohesive and user-friendly design.CSS is instrumental in defining color schemes and theming for the website. In summary, CSS in a music player website is essential for creating an attractive and user-friendly interface, ensuring responsive design, and customizing the appearance of various elements to align with the website's overall theme and branding.

- **Node js:** Node.js is commonly used in the backend of a music player website to handle server-side logic, manage data, and communicate with databases or external APIs.Node.js allows developers to build a server that handles HTTP requests and responses. Node.js is well-suited for developing RESTful APIs that the frontend can interact with. Node.js can handle file uploads and downloads, which is important for managing music files.Node.js can handle user authentication and authorization processes. This includes verifying user credentials, managing user sessions, and ensuring that only authorized users can access certain functionalities.Node.js, with its non-blocking and event-driven architecture, is well-suited for building scalable and performant backend services for music player websites**.**

- **Firebase :** Integrating Firebase into a music player website can enhance functionality, offering features like real-time data synchronization, user authentication, and cloud storage. Use Firebase Realtime Database can be used to store and synchronize music-related data in real time. This includes playlists, track information, and user-specific data. Firebase Authentication can be used to handle user registration and login. This provides secure access to features like personalized playlists and user-specific data. Firebase Cloud Storage is ideal for storing audio files associated with the music tracks. This allows users to upload, retrieve, and stream audio content from the cloud. Firebase Firestore can be used as an alternative to the Realtime Database for structured data storage. Firestore offers powerful querying capabilities and supports more complex data structures. Firebase Hosting can be used to deploy and host the music player website. It provides a secure and scalable hosting solution with features like automatic SSL, CDN, and easy deployment.By integrating Firebase into your music player website, you can leverage a serverless architecture with real-time capabilities, user authentication, and cloud storage, simplifying the development process and enhancing the user experience.

CHAPTER 6 : SYSTEM DESIGN

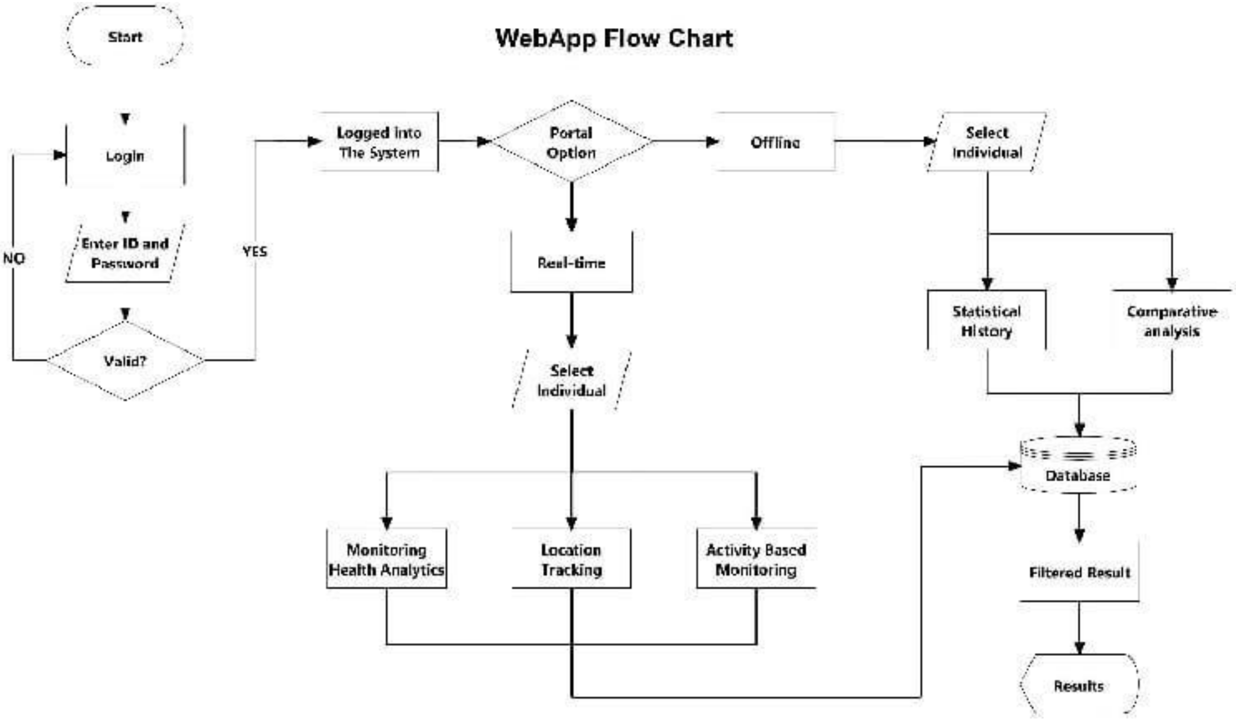


Fig 6.1:Flow chart of front-end.

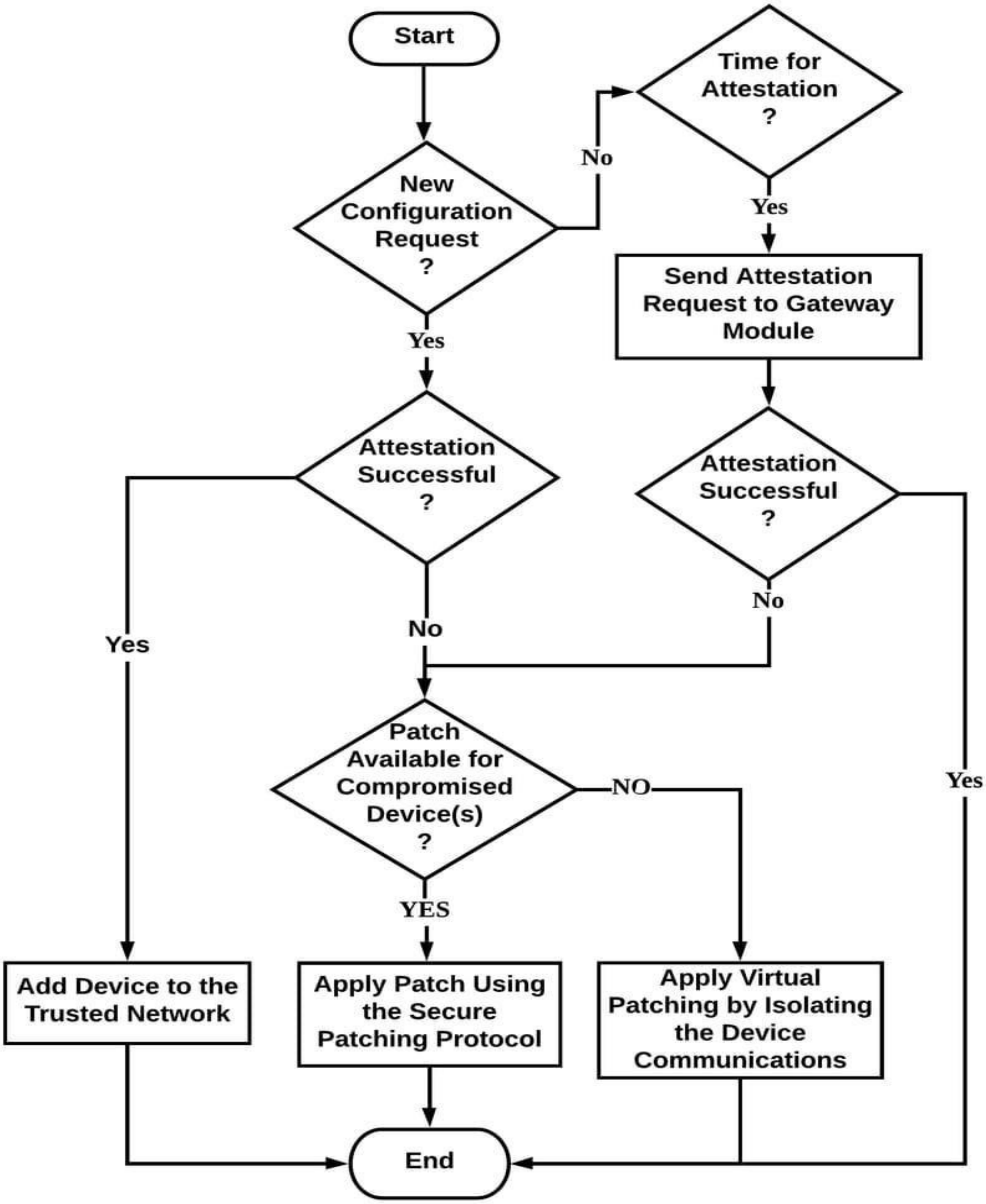

Start
New Configuration Request ?
No
Time for Attestation ?
Yes
Send Attestation Request to Gateway Module
Yes
Attestation Successful ?
Attestation Successful ?
No
Yes
No
Patch Available for Compromised Device(s) ?
NO
Yes
YES
Add Device to the Trusted Network
Apply Patch Using the Secure Patching Protocol
Apply Virtual Patching by Isolating the Device Communications
End

Fig 6.2: Flow chart of back-end.

CHAPTER 7 : IMPLEMENTATION/WORK DONE DURING INTERNSHIP

The initial week of the Fullstack Development internship commenced with a warm welcome and thorough orientation. The team provided an overview of the project's objectives, fostering a sense of inclusion and purpose. Participants engaged in sessions exploring the diverse career opportunities within Fullstack Development. The discussions shed light on the significance of being well-versed in both frontend and backend technologies, emphasizing the role of Fullstack Developers in creating cohesive and efficient software solutions. With a foundational understanding of the internship context and career prospects, hands-on coding sessions commenced. Participants delved into the basics of web development, starting with HTML, CSS, and JavaScript.

7.1 WEEKLY REPORT

7.1.1 WEEK 1:

Introduction, Career Opportunities, Basic Web Development, and Environment Setup

Introduction to the Internship:
The initial week of the Fullstack Development internship commenced with a warm welcome and thorough orientation. The team provided an overview of the project's objectives, fostering a sense of inclusion and purpose.

Career Opportunities in Fullstack Development:
Participants engaged in sessions exploring the diverse career opportunities within Fullstack Development. The discussions shed light on the significance of being well-versed in both frontend and backend technologies, emphasizing the role of Fullstack Developers in creating cohesive and efficient software solutions.

Basic HTML, CSS, and JavaScript:

With a foundational understanding of the internship context and career prospects, hands-on coding sessions commenced. Participants delved into the basics of web development, starting with HTML, CSS, and JavaScript.

Setting up the Development Environment:
An essential aspect of Week 1 was setting up the development environment. Visual Studio Code (VSCode) was chosen as the integrated development environment (IDE) for its versatility and extensive plugin support.

Downloading Visual Studio Code:
Participants were guided through the process of downloading and installing Visual Studio Code from the official website. The step-by-step instructions ensured a smooth setup for both Windows and macOS users.
Visit the [Visual Studio Code website] (https://code.visualstudio.com/).
Click on the "Download" button.
Follow the on-screen instructions for your operating system.
Once the installation is complete, launch Visual Studio Code.

Configuring Visual Studio Code:
To enhance the development experience, participants were introduced to essential VSCode extensions. The "Live Server" extension, for instance, was recommended for quickly launching and previewing HTML files.

- Open Visual Studio Code.
- Navigate to the Extensions view by clicking on the Extensions icon in the Activity Bar on the side of the window or using the keyboard shortcut `Ctrl+Shift+X`.
- Search for "Live Server" and click on the "Install" button.
- Once installed, click on the "Reload" button to activate the extension.
- Open an HTML file, right-click, and select "Open with Live Server" to launch a live preview of the webpage.

7.1.2 WEEK 2:

React Fundamentals and Node.js Installation

Frontend Exploration with React:

The second week of the Fullstack Development internship centered around a comprehensive exploration of React, a JavaScript library for building user interfaces.

Understanding React Components:

Participants delved into the core concept of React components. They grasped the difference between functional and class components and gained a deep understanding of JSX, the syntax extension for JavaScript recommended by React.

State Management in React:

The week extended to the essential topic of state management in React. Participants learned the significance of state in creating dynamic and interactive components and explored the usage of the useState hook.

Hands-On Coding Tasks:

Practical coding tasks were integral to solidifying React concepts. Participants engaged in creating interactive components, handling user input, and managing component state, fostering a hands-on approach to learning.

Introduction to Node.js:

Complementing the frontend exploration, the week introduced participants to the backend aspect of development using Node.js. The event-driven, non-blocking I/O model of Node.js was emphasized, highlighting its suitability for scalable applications.

Node.js Installation:

To enable participants to seamlessly navigate both frontend and backend development, the installation of Node.js was a key focus. Clear instructions were provided for installing Node.js and npm (Node Package Manager) on various operating systems.

1. Visit the [Node.js website](https://nodejs.org/).
2. Download the recommended LTS version for your operating system.
3. Follow the installation instructions provided on the website.
4. Once installed, open a terminal or command prompt and verify the installation using the following commands:
    a. `node -v` (to check Node.js version)
    b. `npm -v` (to check npm version)

7.1.3 WEEK 3:
Login/Signup Page Development and Firebase Integration

Login/Signup Page Development:
Building upon the foundation laid in previous weeks, participants immersed themselves in the development of a login/signup page. The focus was on creating an interactive and user-friendly authentication experience.

Enhancing the Login Page:
The login page structure, introduced in Week 2, was refined and extended to accommodate both login and signup functionalities. Participants implemented input validation and designed a seamless user interface for a streamlined authentication process.

CSS Styling for Authentication Page:
CSS styling was applied to enhance the visual appeal and responsiveness of the login/signup page. Participants fine-tuned the styles to create an engaging user interface.

JavaScript for Authentication Logic:
JavaScript played a pivotal role in implementing the authentication logic. Participants created an auth.js script that handled form submissions, validated user input, and interacted with Firebase for authentication.

Firebase Integration:

The second major aspect of Week 3 was the integration of Firebase for authentication services. Firebase provided a seamless and scalable solution for user authentication.

Setting up Firebase:

Participants were guided through the process of setting up a Firebase project, obtaining API keys, and configuring Firebase Authentication.

Connecting Firebase to the Project:

The Firebase JavaScript SDK was added to the project, enabling seamless integration with Firebase services. Participants learned how to initialize Firebase and access authentication functionalities.

Firebase Configuration:

The configuration details, including the Firebase project's API key, were added to the JavaScript file to establish a connection with Firebase.

7.1.4 WEEK 4:

Working on project

In Week 4 of the Fullstack Development internship, participants embarked on an exciting projects to develop different real-time websites.

Project Objectives:

React Components:

Participants created reusable React components to efficiently manage the UI, ensuring a dynamic and responsive user experience.

API Integration:

The project involved integrating a Movie Database API to fetch and display relevant movie information. Axios, a promise-based HTTP client, facilitated seamless communication with the API.

7.2 PROJECT

7.2.1 OVERVIEW:

We completed a project that involved creating a basic music player website using HTML, CSS, and JavaScript. This project was designed to showcase our skills in front-end web development. It is a responsive and user-friendly web application that allows users to play music. We used HTML and CSS to design and style the interface, and We implemented JavaScript to handle user input and interaction such as basic playback controls -play, pause, and next. We also gained experience working with version control tools such as Git and GitHub.

7.2.2 FEATURES:

- Play/Pause: Allows the user to start or pause the playback of the currently selected track.
- Next/Previous: Enables the user to skip to the next or previous track in the playlist or album.
- Seek: Allows the user to manually move forward or backward within a track by dragging a slider or clicking on a progress bar.
- Display the currently playing song and album artwork.

7.2.3 USAGE:

Once the Music player site is open in your browser, You can then click the play button next to the song to start playing it, and use the playback controls at the bottom of the page to control the song playback. The currently playing song and album artwork are displayed in the "Now Playing" section at the bottom of the page.

7.3 CODE:

7.3.1 HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Music PLayer - Change Your Tune</title>
    <link rel="stylesheet" href="style.css" />
    <link rel="icon" type="image/x-icon" href="icon.jpeg" />
  </head>
  <body>
    <nav>
      <ul>
        <li class="brand"><img src="logo.jpeg" alt="Music Player" /> Music Player</li>
      </ul>
    </nav>
    <div class="container">
      <div class="songList">
        <h1>Trending Now!</h1>
        <div class="songItemContainer">
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Ayyo Vayye</span>
            <span class="songlistplay"><span class="timestamp">03:46
                <i id="0" class="far songItemPlay fa-play-circle"></i> </span ></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Idhazhin Oram</span>
            <span class="songlistplay"><span class="timestamp">03:25
                <i id="1" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Ordinary Person</span>
            <span class="songlistplay"><span class="timestamp">02:18
<i id="2" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Kahani Suno</span>
            <span class="songlistplay"><span class="timestamp">02:49
                <i id="3" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Hold On </span>
            <span class="songlistplay"><span class="timestamp">03:08
                <i id="4" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Night Changes</span>
            <span class="songlistplay"><span class="timestamp"  >03:51
                <i id="5" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Rewrite The Stars</span>
            <span class="songlistplay"><span class="timestamp">03:39
                <i id="6" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Perfect</span>
            <span class="songlistplay"><span class="timestamp">04:23
                <i id="7" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">In The Stars</span>
            <span class="songlistplay"><span class="timestamp">03:36
                <i id="8" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
          <div class="songItem">
            <img alt="1" />
            <span class="songName">Iraadhay</span>
            <span class="songlistplay"><span class="timestamp">02:13
                <i id="9" class="far songItemPlay fa-play-circle"></i> </span></span>
          </div>
        </div>
      </div>
      <div class="songBanner"></div>
    </div>
    <div class="bottom">
      <input
        type="range"
        name="range"
        id="myProgressBar"
        min="0"
        value="0"
        max="100"
      />
      <div class="icons">
        <i class="fas fa-3x fa-step-backward" id="previous"></i>
        <i class="far fa-3x fa-play-circle" id="masterPlay"></i>
        <i class="fas fa-3x fa-step-forward" id="next"></i>
      </div>
      <div class="songInfo">
        <img src="playing.gif" width="42px" alt="" id="gif" />
        <span id="masterSongName">Ayyo Vayye</span>
      </div>
    </div>
    <script src="script.js"></script>
    <script
      src="https://kit.fontawesome.com/26504e4a1f.js"
      crossorigin="anonymous"
    ></script>
```

```
  </body>
</html>
```

7.3.2 JS CODE:

```
let songIndex = 0;
let audioElement = new Audio('songs/1.mp3');
let masterPlay = document.getElementById('masterPlay');
let myProgressBar = document.getElementById('myProgressBar');
let gif = document.getElementById('gif');
let masterSongName = document.getElementById('masterSongName');
let songItems = Array.from(document.getElementsByClassName('songItem'));
let songs = [
{songName: "Ayyo Vayye", filePath: "songs/1.mp3", coverPath: "covers/1.jpeg"},
{songName: "Idhazhin Oram", filePath: "songs/2.mp3", coverPath: "covers/2.jpeg"},
{songName: "Ordinary Person", filePath: "songs/3.mp3", coverPath: "covers/3.jpeg"},
{songName: "Kahani Suno", filePath: "songs/4.mp3", coverPath: "covers/4.jpeg"},
{songName: "Hold On", filePath: "songs/5.mp3", coverPath: "covers/5.jpeg"},
{songName: "Night Changes", filePath: "songs/6.mp3", coverPath: "covers/6.jpeg"},
{songName: "Rewrite The Stars", filePath: "songs/7.mp3", coverPath: "covers/7.jpeg"},
{songName: "Perfect", filePath: "songs/8.mp3", coverPath: "covers/8.jpeg"},
{songName: "In The Stars", filePath: "songs/9.mp3", coverPath: "covers/9.jpeg"},
{songName: "Iraadhay", filePath: "songs/10.mp3", coverPath: "covers/10.jpeg"},
]
songItems.forEach((element, i)=>{
element.getElementsByTagName("img")[0].src = songs[i].coverPath;
element.getElementsByClassName("songName")[0].innerText  =  songs[i].songName;
})
masterPlay.addEventListener('click', ()=>{
if(audioElement.paused || audioElement.currentTime<=0){
audioElement.play();
masterPlay.classList.remove('fa-play-circle');
masterPlay.classList.add('fa-pause-circle');
gif.style.opacity = 1;
}
else{
audioElement.pause();
masterPlay.classList.remove('fa-pause-circle');
masterPlay.classList.add('fa-play-circle');
gif.style.opacity = 0;
}})
audioElement.addEventListener('timeupdate',  ()=>{
progress = parseInt((audioElement.currentTime/audioElement.duration)* 100);
myProgressBar.value = progress;
})
myProgressBar.addEventListener('change',  ()=>{
audioElement.currentTime = myProgressBar.value * audioElement.duration/100;
})
const makeAllPlays = ()=>{
Array.from(document.getElementsByClassName('songItemPlay')).forEach((element)=>{
```

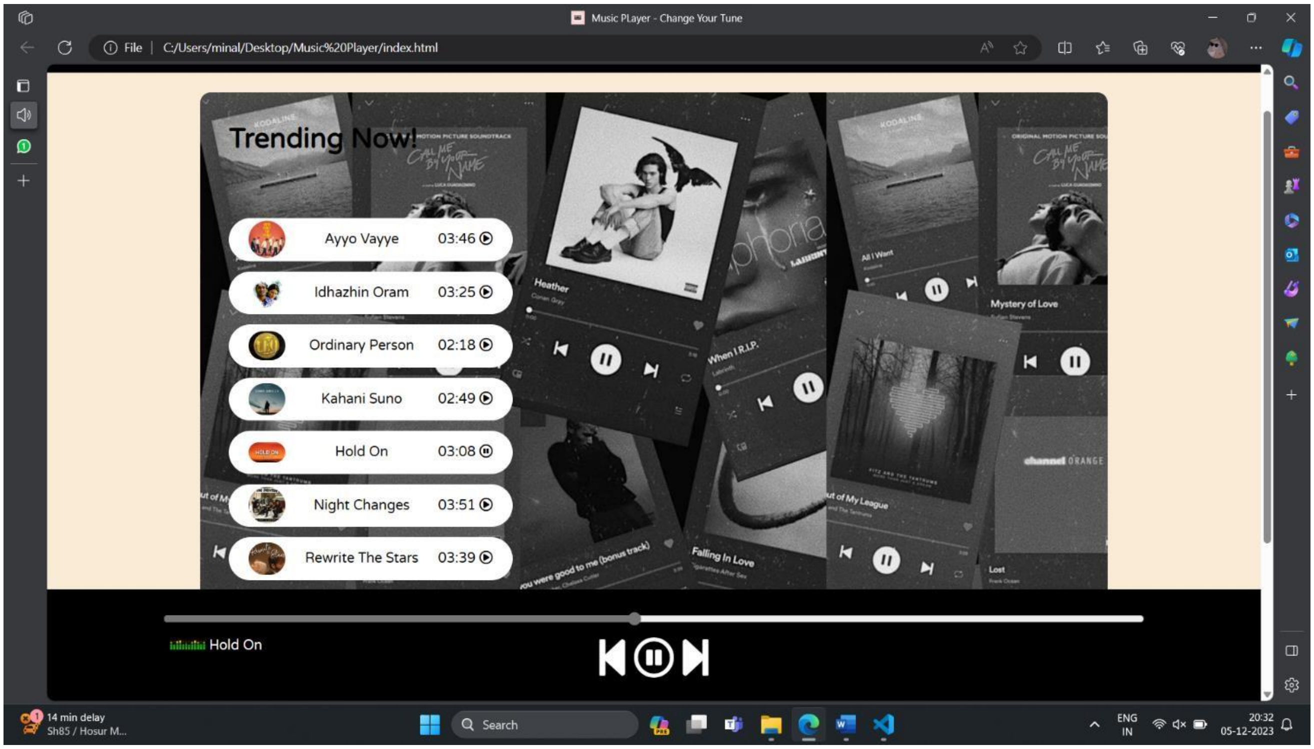
```
element.classList.remove('fa-pause-circle');
element.classList.add('fa-play-circle');
})}
Array.from(document.getElementsByClassName('songItemPlay')).forEach((element)=>{
element.addEventListener('click', (e)=>{
makeAllPlays();
songIndex = parseInt(e.target.id);
e.target.classList.remove('fa-play-circle');
e.target.classList.add('fa-pause-circle');
audioElement.src = `songs/${songIndex+1}.mp3`;
masterSongName.innerText = songs[songIndex].songName;
audioElement.currentTime = 0;
audioElement.play();
gif.style.opacity = 1;
masterPlay.classList.remove('fa-play-circle');
masterPlay.classList.add('fa-pause-circle');
})})
document.getElementById('next').addEventListener('click', ()=>{
if(songIndex>=9){
songIndex = 0
}
else{
songIndex += 1;
}
audioElement.src = `songs/${songIndex+1}.mp3`;
masterSongName.innerText = songs[songIndex].songName;
audioElement.currentTime = 0;
audioElement.play();
masterPlay.classList.remove('fa-play-circle');
masterPlay.classList.add('fa-pause-circle');
})
document.getElementById('previous').addEventListener('click', ()=>{
if(songIndex<=0){
songIndex = 0
}
else{
songIndex -= 1;
}
audioElement.src = `songs/${songIndex+1}.mp3`;
masterSongName.innerText = songs[songIndex].songName;
audioElement.currentTime = 0;
audioElement.play();
masterPlay.classList.remove('fa-play-circle');
masterPlay.classList.add('fa-pause-circle');
})
```

7.3.3 CSS CODE:

```
@import url('https://fonts.googleapis.com/css2?family=Ubuntu&display=swap');
@import url('https://fonts.googleapis.com/css2?family=Varela+Round&display=swap');
```

```
body{
background-color:  antiquewhite;}
*{
margin: 0;
padding: 0;}
nav{
font-family: 'Ubuntu', sans-serif;}
nav ul{
display: flex;
align-items: center;
list-style-type: none;
height: 65px;
background-color: black;
color: white;}
.brand img{
width: 44px;
padding: 0 8px;}
.brand {
display: flex;
align-items:  center;
font-weight: bolder;
font-size: 1.3rem;}
.container
{
min-height: 72vh;
background-color: black;
color: black;
font-family: 'Varela Round', sans-serif;
display: flex;
margin: 23px auto;
width: 70%;
border-radius: 12px;
padding: 34px;
background-image:  url('bg.jpeg');}
.bottom
{
position: sticky;
bottom: 0;
height: 130px;
background-color: black;
color: white;
display: flex;
justify-content: center;
align-items: center;
flex-direction: column; }
.icons{
margin-top: 14px; }
.icons i{
cursor: pointer;}
#myProgressBar{
width: 80vw;
cursor: pointer;}
.songItemContainer{
margin-top: 74px;}
.songItem
{
height: 50px;
display: flex;
background-color: white;
color: black;
margin: 12px 0;
justify-content: space-between;
align-items: center;
border-radius: 34px;}
.songItem img{
width: 43px;
margin: 0 23px;
border-radius: 34px;}
.timestamp{
margin: 0 23px;}
.timestamp i{
cursor: pointer;}
.songInfo{
position: absolute;
left: 10vw;
font-family: 'Varela Round', sans-serif;
}
.songInfo img{
opacity: 0;
transition: opacity 0.4s ease-in;
}
```

7.4 SNAPSHOT:



Music PLayer - Change Your Tune
File | C:/Users/minal/Desktop/Music%20Player/index.html
Trending Now!
Hold On
14 min delay
Sh85 / Hosur M...
Search
ENG
IN
20:32
05-12-2023

Ayyo Vayye	03:46
Idhazhin Oram	03:25
Ordinary Person	02:18
Kahani Suno	02:49
Hold On	03:08
Night Changes	03:51
Rewrite The Stars	03:39

CHAPTER 8 : REFLECTION/LEARNINGS

The culmination of the Fullstack Development internship marks a journey filled with growth, challenges, and profound learning experiences. Throughout the internship, participants engaged in a progressive exploration of frontend and backend technologies, honing their skills to become adept developers.

Key Learnings:

Frontend Mastery:

Proficiency in creating dynamic and visually appealing user interfaces using HTML, CSS, and JavaScript.

In-depth understanding and application of React for building interactive components.

Backend Proficiency:

Introduction to backend development with Node.js, providing a comprehensive Fullstack skill set.

Integration of external services, such as Firebase, for robust authentication.

API Integration and Project Development:

Practical experience in working with external APIs, demonstrated in the creation of an image searching website.

Development of real-world projects, translating theoretical knowledge into tangible applications.

CHAPTER 9 : CONCLUSION

The music player website aims to provide a seamless and enjoyable music listening experience. With its user-friendly interface, diverse and customizable playlist, it caters to a wide audience. Additionally, features like personalized recommendations and offline listening enhance user satisfaction, making it a comprehensive platform for music enthusiasts. Constant updates and user feedback integration ensure the website stays dynamic and aligned with evolving preferences in the ever-changing world of music.

Full-stack web development is a dynamic and integral discipline that encompasses both front-end and back-end technologies, enabling developers to create comprehensive and interactive web applications. In this multifaceted role, professionals proficient in languages like HTML, CSS, JavaScript, and server-side frameworks navigate the entire development stack, ensuring seamless user experiences and robust functionality.

The significance of full-stack expertise lies in its versatility, allowing developers to adapt to evolving industry trends and demands. By mastering both client and server-side aspects, developers can contribute holistically to project lifecycles, fostering efficient collaboration and faster development cycles. This integrated skill set is particularly valuable in startups and smaller teams where individuals wear multiple hats.

Moreover, the continuous evolution of full-stack technologies underscores its enduring relevance. The demand for full-stack developers remains high as businesses seek agile and comprehensive solutions. In conclusion, embracing full-stack web development empowers professionals to build end-to-end solutions, stay adaptable in a dynamic tech landscape, and play pivotal roles in shaping the digital experiences of tomorrow.

CHAPTER 10 : REFERENCES

- Wikipedia
- Geeksforgeeks
- Pinterest