

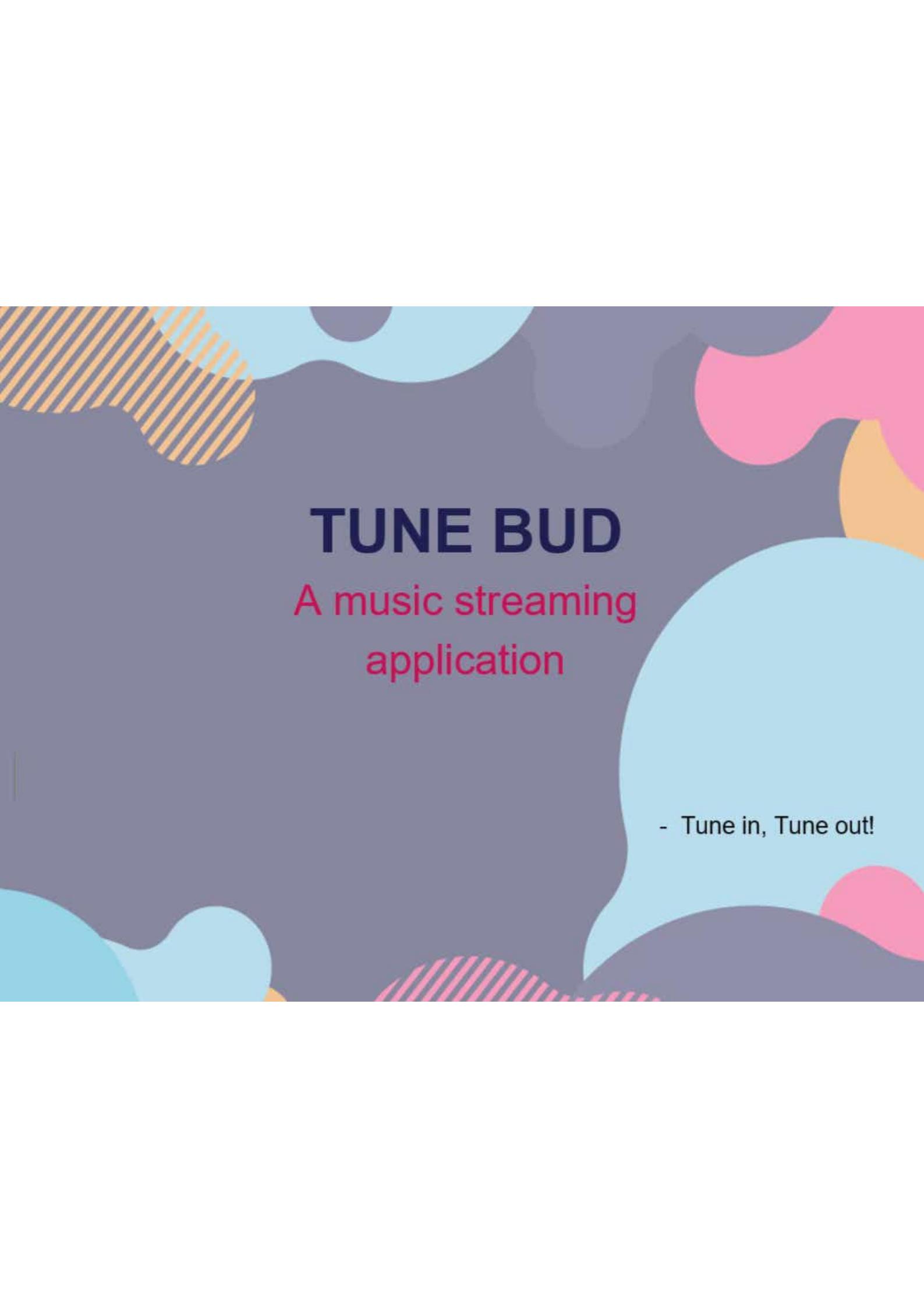
Full Stack with MERN

Through SmartInternz
via
APSCHE

DR. Lankapalli Bullayya College

TEAM MEMBERS

1. **Joshna Rani Buthada**
2. **Akash Vaddi Dhanaraju**
3. **Allada Pavan Kumar**
4. **Loddu Upendra**



TUNE BUD

A music streaming
application

- Tune in, Tune out!

I. INTRODUCTION :-

(a) Over View:- Tune Bud is a cutting-edge music streaming application designed to deliver a seamless and immersive musical experience to users worldwide. With its intuitive interface, vast library of songs, and innovative features, Tune Bud aims to revolutionize the way people listen to and discover music.

Extensive music library:- Tune Bud boasts a vast collection of songs spanning multiple genres, artists, and languages, ensuring that users can find their favourite tunes and discover new ones effortlessly.

Personalised Recommendations:- Using advanced algorithms, Tune Bud analyzes user's listening habits, preferences, and history to offer personalized song recommendations, curated playlists, and tailored radio stations, enhancing the discovery of new music.

High quality Audio streaming! With support for high-definition audio-streaming, Tune Bud ensures that users enjoy crystal-clear sound quality, allowing them to immerse themselves fully in music.

(B) Purpose:-

The purpose of the tune bud music streaming application is to provide users with a comprehensive and enjoyable music listening experience, characterized by convenience, personalized recommendations, high-quality audio, community engagement, offline accessibility, and cross-platform compatibility.

Convenience & Accessibility: Tune Bud aims to provide users with a convenient and accessible platform to enjoy their favorite music anytime, anywhere.

Enhanced Music Discovery: By leveraging advanced algorithms and personalized recommendations, Tune Bud seeks to enhance users' music discovery experience.

Quality and Immersion: Tune Bud prioritizes delivering high-quality audio streaming to ensure users enjoy an immersive and engaging listening experience.

Community Building: Beyond just a music streaming platform, Tune Bud aims to foster a vibrant and interactive music community.

Offline Enjoyment: Recognizing the importance of accessibility in all situations, Tune Bud allows users to download their favorite songs, albums, and playlists for offline listening.

2. LITERATURE SURVEY

a) Existing problem :-

One existing problem that Tune Bud addresses is the fragmentation and inconvenience often associated with traditional music consumption methods. Before, the advent of music streaming services like Tune Bud, users often relied on multiple platforms such as CDs, digital downloads, or various streaming services, to access their favorite music. This fragmented approach not only led to cluttered music libraries but also required users to manage multiple accounts and subscriptions, resulting in a disjointed and less-than-optimal user experience.

Additionally, traditional methods of music consumption often lacked personalized recommendations and discovery features, making it challenging for users to explore new artists & genres that aligned with their tastes. Users were limited to their existing knowledge & had to rely on external sources for recommendations, which could be time-consuming and hit-or-miss.

(b) Proposed Solution:-

The proposed solution offered by Tune Bud is designed to address the existing problems in music consumption methods by providing a comprehensive and user-centric platform. Here's how Tune Bud's features serve as a solution.

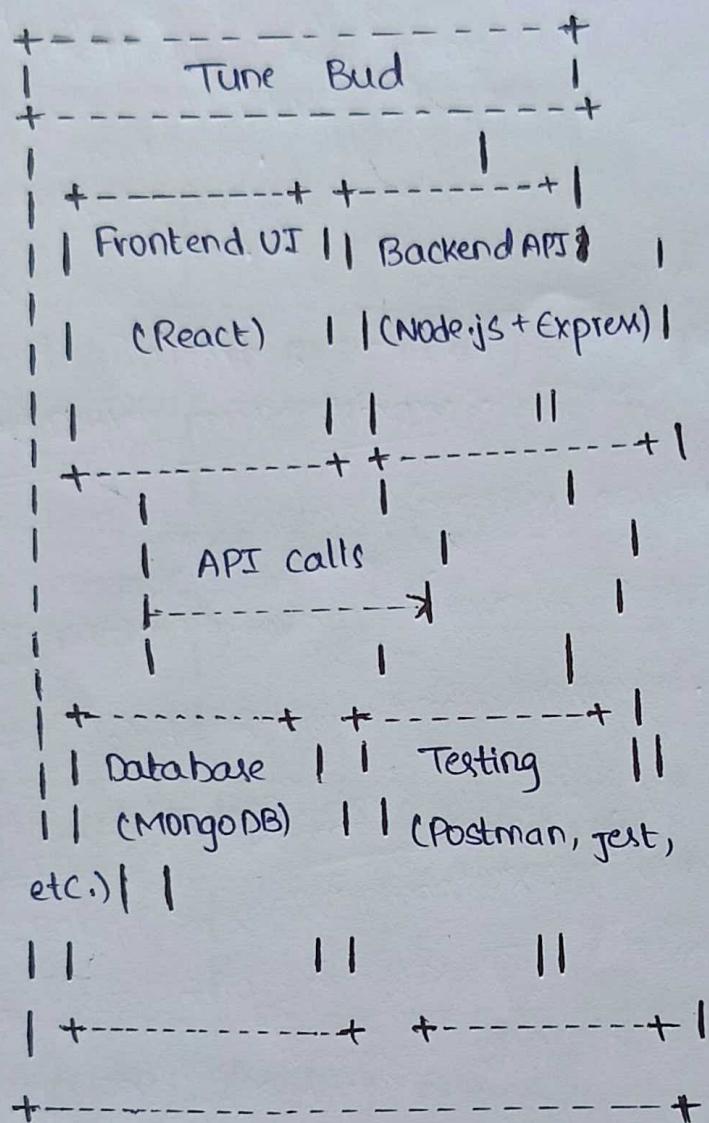
Unified Platform: Tune Bud consolidates various music consumption methods into one unified platform, eliminating the need for users to juggle between CDs, digital downloads, or multiple streaming services. By offering a vast library of songs across genres and languages, Tune Bud becomes the go-to destination for all music-related needs.

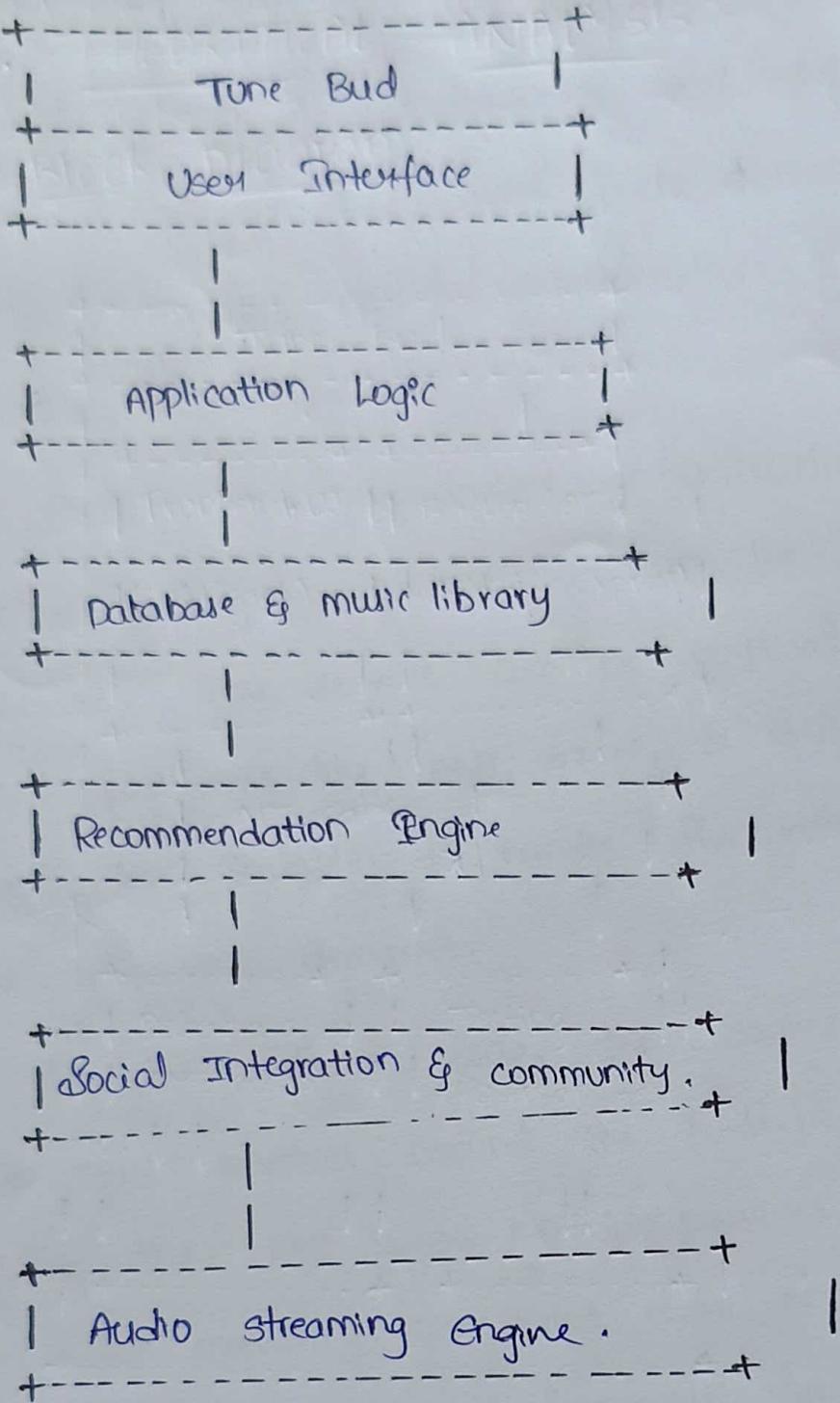
Personalized Recommendations: Leveraging advanced algorithms and user data analysis, Tune Bud delivers personalized song recommendations, curated playlists, and tailored radio stations.

Social Integration: Tune Bud integrates social features that enable users to connect with friends, share their favorite tracks, playlists, and discoveries, and engage in music-related discussions.

3. THEORITICAL ANALYSIS :-

(a) Block Diagram:-





(b) Hardware/ Software design)

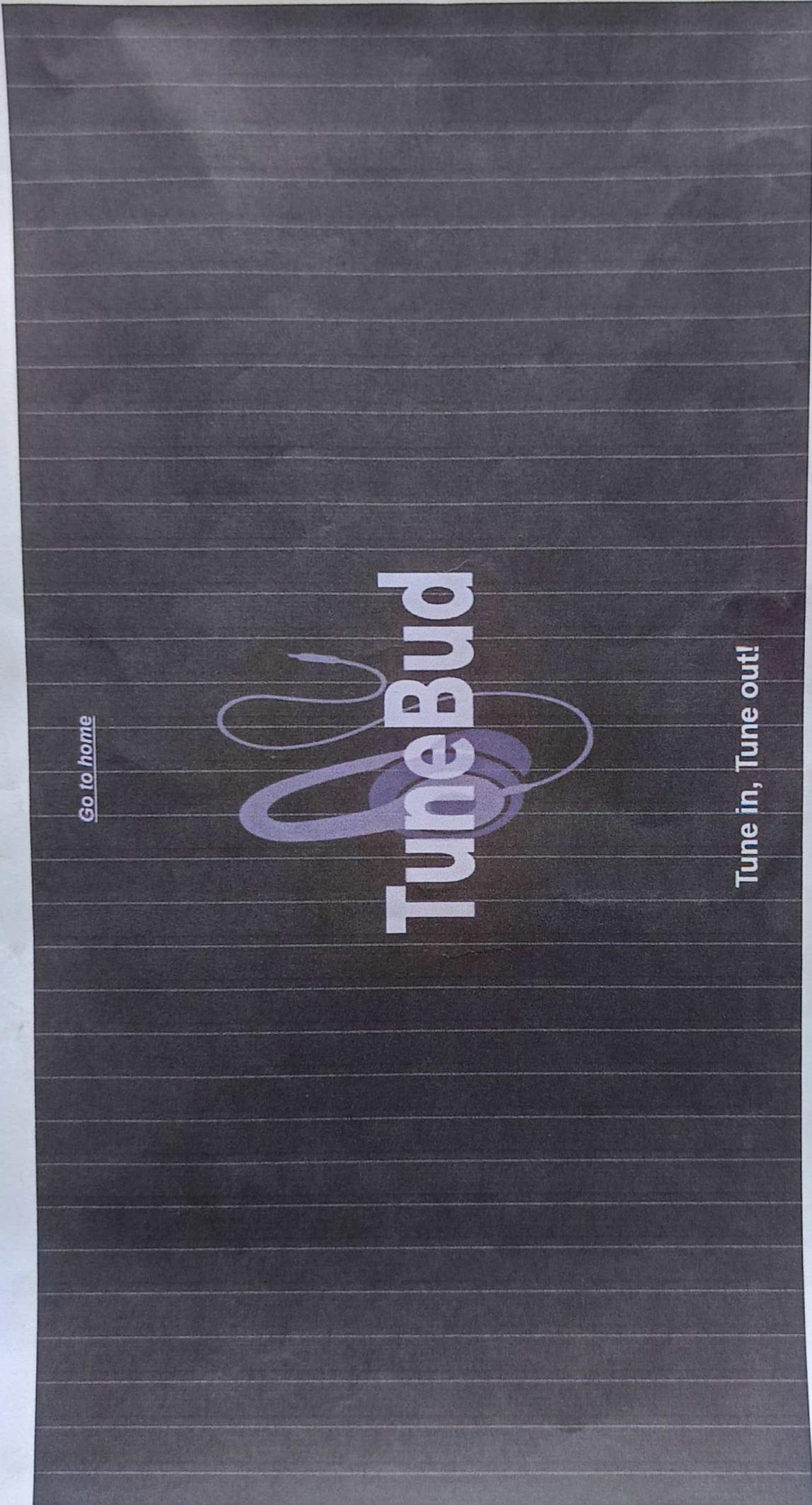
Designing the hardware and software architecture for a project like Tune Bud, a music streaming application, requires careful consideration of various factors to ensure its success.

Firstly, the hardware infrastructure for Tune Bud involves setting up servers to host the backend API and database. These servers can be physical machines or cloud-based platforms like AWS, Azure, or GCP. The infrastructure should be scalable to handle fluctuating user traffic & data processing demands.

On the software side, the Backend architecture of Tune Bud revolves around the Node.js and Express.js for building the API layer. Node.js provides a runtime environment for executing Javascript code on the server-side, while Express.js offers a framework for building RESTful APIs and defining routes.

For the frontend architecture, React is the library of choice for building reusable UI components and managing component state.

(4) RESULT:





To continue, log in to Jo Tunes.

Email address or userName

Email address or userName

Password

Password

LOGIN

Don't have an account?

SIGN UP FOR SPOTIFY



Sign up to start listening

First Name

Enter your first name.

Last Name

Enter your last name.

username

Enter your username

Email address

domain@gmail.com,

Confirm Email Address

Confirm Email Address.

Create Password

Enter a strong password

Sign Up



Tune in, Tune out!



Home

Search

Library

My Music

Create Playlist

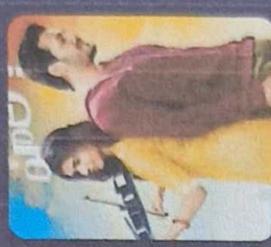
Liked Songs



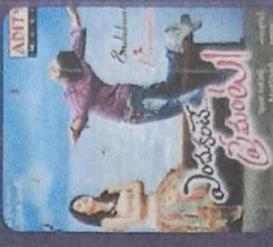
vellipove vellipove



Nuvvu navvukuntu
vallinomale



Anaganaga oka uru



Nee chupule



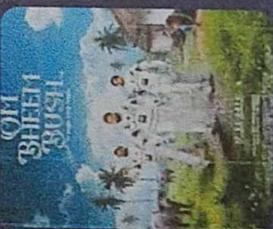
Adhey name



Nijame ne chebutihunna



Oy...Oy



anuvanuvu



Kannulu basalu



Poolamme pilla



Premisthunnu



Chali gaali Chuuddu



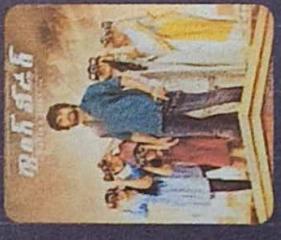
Kahani vacha vacha



Adiga adiga



Megham Karigenu



Hoyna hoyna



Home

Search

Library

My Music

Create Playlist

Liked Songs

English



Upload Song

Download

Support

Premium

Q What do you want to listen to?

Nothing to show!

Home

Search

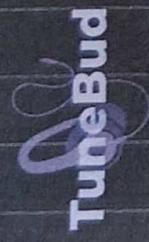
Library

My Music

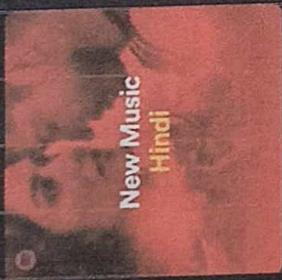
Create Playlist

Liked Songs

English

[Upload Song](#)[Download](#)[Support](#)[Premium](#)

My Playlists

[Hindi](#)[Create Playlist](#)[Liked Songs](#)[Home](#)[Search](#)[Library](#)[My Music](#)

English

My Songs



Oy....Oy
Siddarth



Adhey nanne
Karthik, V Prasanna



Anaganaga oka uru
Sn Dhruvi



vellipove vellipove
Ranithi



Anuvanuvu
Sunny M R, Arijit Singh



Nijame ne chebutihunna
Sid Sriram



Nee chupule
K S Chitra, Han Charan



Nuvvu navvukuntu vellipomamake
Kapil Kapilan



Megham kairigenu
Anushka, Venkatesh, Latha



Oy....Oy
Siddarth

4:42

5:23

5:29

4:38

3:31

3:56

6:02

3:24

4:50

0:27



My Songs



Oy



Adhey name
Kannada Prashant



Anaganaga ok
Sathish



vellipove vellip
Dhan



Anuvacuvu
Samay H.R. A.R.S.



Njame ne che
Shreya Ghoshal



Nee chupule
Gaura Hanika



Nuvvu navvukuntu vellipomake
Karthik S. Pillai



Megham kari genu
Karthik S. Pillai



Kalyani vacha vacha
Karthik S. Pillai

TuneBud

Home

Search

Library

My Music

Create Playlist

Liked Songs

English

Upload Song

Download

Support

Premium

User

Upload Your Music

Name

Singer(s)

Duration

Name

Singer(s)

duration

Thumbnail

Select Track

Submit Song

(5) ADVANTAGES AND DISADVANTAGES

Advantages :-

Scalability: Cloud-based infrastructure allows for easy scalability to accommodate growing user traffic and data demands without significant upfront investment in hardware.

Reliability: Redundant server configurations and load balancers ensure high availability and fault tolerance, minimizing downtime and ensuring uninterrupted service for users.

Cost-effectiveness: Cloud-based infrastructure eliminates the need for upfront capital expenditure on hardware and allows for pay-as-you-go pricing models, optimizing costs based on usage.

Flexibility: Modular architecture allows for easy addition & modification of components, enabling rapid development and deployment of new features and updates.

Performance: High-speed internet connectivity and optimized backend and frontend components ensure fast response times and smooth user experience, even during peak usage periods.

Disadvantages:

Dependency on Internet connectivity: The reliance on internet connectivity for accessing the application may result in limited functionality & usability in areas with poor or unstable internet connections.

Complexity: Managing a cloud-based infrastructure and co-ordinating multiple software components can be complex and requires expertise in system administration and software engineering.

Vendor Lock-in: Using cloud-based services may result in vendor lock-in, making it challenging to migrate to alternative platforms or providers in the future.

Cost Over-runs: While cloud-based services infrastructure offers cost-effectiveness, inefficient resource utilization & unexpected usage spikes can lead to cost overruns if not carefully monitored and managed.

Security Risks: Cloud-based environments may be susceptible to security breaches & data leaks if proper security measures are not implemented & maintained.

Performance Variability: The performance of cloud-based services may vary depending on factors such as server load, network congestion, and geographical location, affecting the overall user experience unpredictably.

(6) APPLICATIONS:

The solution of designing a music streaming application like Tune Bud can be applied in various areas where there is a demand for digital music consumption and exploration. Here are some key areas where this solution can be effectively applied.

Entertainment Industry: Music streaming applications cater to the entertainment needs of millions of users worldwide.

Media and Broadcasting companies: This can integrate music streaming applications into their digital platforms to offer additional services to their audience.

Education and Learning platforms: Music streaming applications can be integrated into education and learning platforms to provide access to educational content such as music theory lessons, instrument tutorials and historical recordings.

Fitness and well-being apps: This can incorporate music streaming features to provide users with curated playlists and personalized music recommendations for workout sessions, meditation or relaxation.

Mobile and Smart Devices: Music streaming applications can be pre-installed or integrated into mobile devices, smart speakers, and other IoT (Internet of things) devices to offer seamless access to music on the go. This enhances the value proposition of such devices and enhances user satisfaction.

(7) CONCLUSION :-

In conclusion, the solution of designing a music streaming application like Tune Bud offers a versatile platform that can be applied across various industries and sectors to meet diverse needs and objectives. By providing users with convenient access to a vast library of songs, personalized recommendations, and interactive features, music streaming applications enhance entertainment experiences, foster community engagement, and support educational and wellness initiatives.

The flexibility and scalability of such applications make them suitable for integration into digital platforms, entertainment services, education platforms, retail establishments, event management services and more. Whether it's enhancing customer experiences in retail stores, providing educational content in learning platforms, or fostering community engagement on social networking platforms, music streaming applications offer a valuable tool to connect with audiences, drive management engagement, and achieve business objectives.

Furthermore, the adoption of cloud-based infrastructure and modern software development practices enables rapid

development, deployment, and iteration of music streaming applications, ensuring continuous improvement and innovation. By leveraging emerging technologies, data analytics, and user feedback, organizations can optimize their music streaming offerings to deliver personalized, engaging, and memorable experiences to users worldwide.

In summary, the solution of designing a music streaming application presents a compelling opportunity for organizations to enhance user experiences, drive engagement, and achieve their objectives across various industries and sectors. As technology continues to evolve, music streaming applications will remain a valuable tool for connecting with audiences, fostering community engagement, and delivering value in the digital age.

(8) FUTURE scope :-

The Future scope for music streaming applications like Tune Bud is promising, with opportunities for innovation and growth in various areas. Here are some key aspects of the future scope for such applications.

Enhanced Personalization: Future music streaming applications will leverage advanced data analytics, machine learning, and artificial intelligence to offer even more personalized recommendations and curated playlists.

Immersive User Interfaces: With advancements in technology such as augmented reality (AR) and virtual reality (VR), music streaming applications could offer immersive user interfaces that provide a more engaging and interactive experience.

Integration with smart Devices:

Music streaming applications will continue to integrate with a wide range of smart devices, including smart speakers, wearables and connected cars.

Global Expansion: Music streaming applications will continue to expand their reach to new markets and regions worldwide, catering to diverse cultural preferences and languages.

Content Diversity: Future music streaming applications will offer a broader range of content beyond traditional music, including podcasts, audio books, live recordings and exclusive artist content.

ACTIVITY LOG FOR THE FIRST WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day - 1	Introduction to full stack, frontend, Backend. Roles & Responsibility for full stack Development Environment Setup.	Introduction to full stack, you will grasp the concept & significance of full stack development while delving into the essential technologies & frameworks.	
Day - 2	HTML Fundamentals, Introduction to HTML, Basic HTML structure, HTML elements, HTML attributes, HTML lists, HTML Images.	Learn HTML Basics, Understand HTML's purpose, syntax, & role in building the foundation of web pages.	
Day - 3	HTML Fundamentals, HTML forms, HTML Tables, HTML units, HTML Layout.	Master HTML basics, forms, tables, links & layout for structured web content.	
Day - 4	CSS Basics, CSS Selectors, CSS properties, Box model.	Master CSS basics, Selectors, properties & the box model for styling web pages. Learn to apply CSS to select elements.	
Day - 5	CSS Layouts, CSS Grid, CSS Flexbox.	Master CSS layouts, Grid, and Flexbox for advanced web design. Learn to create responsive layouts using CSS Grid & Flexbox for modern web applications.	by Nitika
Day - 6	Introduction to JavaScript, Variables, datatypes, & operators, Control flow statements, ES6, Synchronous & Asynchronous Programming.	Master JavaScript basics, including variables, datatypes, operators, & control flow statements for effective programming. Learn ES6 features.	

WEEKLY REPORT

WEEK - 1 (From Dt..... to Dt.....)

Objective of the Activity Done:

Detailed Report:

INTRODUCTION TO FULL STACK DEVELOPMENT

- **CONCEPT:** Full stack development involves working on both the front-end and backend of web applications.
- **SIGNIFICANCE:** Full stack developers can independently work on all aspects of a web project, leading to faster development cycles and better collaboration.

FRONTEND DEVELOPMENT:

- **HTML Fundamentals:** Basic structure, elements, attributes, lists, and images are key components of HTML.
- **CSS Basics:** Selectors, properties, and the box model are fundamental to CSS.
- **JavaScript:** Covers variables, data types, operators, control flow statements (if-else, for, while, switch), and ES6 features (let, const, template strings).

ENVIRONMENT SETUP:

- Setting up a full stack development environment involves installing and configuring tools and frameworks for frontend (e.g., IDEs, browsers, CSS preprocessors) and backend (e.g., databases, servers, APIs).

ACTIVITY LOG FOR THE SECOND WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day - 1	Arrow function, Spread operator, destructuring, callback promise, Java script fundamentals: functions Objects, arrays Manipulating the DOM with Javascript handling events & user interactions with JavaScript.	Master key Java script concepts & fundamentals Gain proficiency in DOM manipulation & event handling for interactive web development.	J
Day - 2	Introduction to React & its features setting up a React development environment JSX syntax & its Benefits.	Master React basics, including its features & JSX syntax, & learn to setup a development environment for React projects.	
Day - 3	React components, Props and conditional Rendering creating React components. creating reusable React components using props to pass data b/w components.	Learn to create & reuse React components, utilize props for inter-component data transfer, & implement conditional rendering for dynamic user interfaces.	
Day - 4	React state, Event Handling & forms creating conditional rendering & handling events in React Understanding state & its importance in React Setting state & handling events in React.	Master React state management, event handling, & form implementation, crucial for dynamic user interactions & responsive applications.	
Day - 5	Using forms & controlled components in React Handling errors & edge cases in React.	Become proficient in using forms & controlled components in React, mastering error handling & edge case management for resilient applications.	
Day - 6	React Lifecycle & Hooks Understanding the React lifecycle & its phases Using lifecycle methods to optimize performance Introduction to React hooks Implementing custom hooks in React.	Master React's lifecycle phases & methods for optimizing performance, & learn about React hooks, including custom hook implementation for efficient functional component logic.	Af. NHT

WEEKLY REPORT

WEEK - 2 (From Dt..... to Dt.....)

Objective of the Activity Done:

Detailed Report:

Master key Java script concepts and React fundamentals, including state management, component creation, event handling, and hooks, for building interactive and efficient web applications. Gain proficiency in DOM manipulation, form handling, error management, and edge case handling in React for robust and responsive web development.

ACTIVITY LOG FOR THE THIRD WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day - 1	Introduction to Node.js & its features understanding event-driven programming setting up the development environment Installing Node.js & npm setting up a basic project structure	Master Node.js fundamentals, learn to setup a development environment with Node.js & npm, including a basic project structure for Node.js applications.	
Day - 2	Core modules (fs, http, etc.) commonJS modules and npm packages Node events handling asynchronous operations.	Become a proficient in core Node.js modules such as fs and http, understand commonJS modules for file system, mail & URL operations.	
Day - 3	Introduction to express.js setting up an express application Routing & middleware concepts Handling HTTP requests & responses Creating routers & parsing request data & sending responses.	Master Express.js fundamentals, including setting up an application, understanding routing & middleware concepts and handling HTTP requests.	
Day - 4	Building a webserver with express.js route parameters & query strings handling dynamic routes extracting data from URL parameters and query strings.	Master building web servers with express.js, including handling route parameters and query strings, & dynamically extracting data from URL.	
Day - 5	Building RESTful APIs with Node & express.js Understanding the principles of RESTful APIs Designing RESTful APIs RESTful principles & best practices.	Master building RESTful APIs with Node.js & express.js, including understanding the principles & best practices of RESTful API design.	
Day - 6	Building RESTful APIs with Node & express.js CRUD operations with REST Building RESTful APIs with express creating API endpoints.	Master building RESTful APIs with Node.js & express.js, including CRUD operations.	

G. M. H.

WEEKLY REPORT

WEEK - 3 (From Dt..... to Dt.....)

Objective of the Activity Done:

Detailed Report:

INTRODUCTION TO Node.js & its FEATURES

- Learn about Node.js and its features for building scalable and efficient server-side applications.
- Understand event-driven programming; a key paradigm in Node.js for handling synchronous operations.

Setting Up the Development Environment :-

- Install Node.js and npm to set up a basic project structure for Node.js applications.
- Get familiar with the tools and processes needed to start developing with Node.js.

Core modules and commonJS Modules :-

- Explore core Node.js modules like fs (file system) & http for handling file operations and creating web servers.
- Understand commonJS modules for organizing code and leveraging npm packages for enhanced functionality.

Node events and Asynchronous Operations

- Learn to handle asynchronous operations efficiently, a crucial aspect of Node.js development.

ACTIVITY LOG FOR THE FORTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day - 1	Introduction to MongoDB understanding NoSQL databases and MongoDB development environment. Building MongoDB schema and models with Mongoose to perform CRUD operations in MongoDB.	Master MongoDB basics, learn to build schemas, models, & perform CRUD operations using Mongoose for efficient data management.	
Day - 2	Implementing authentication and authorization with MongoDB. Indexes in MongoDB. Importance of indexes creating and managing indexes.	Master implementing authentication & authorization using MongoDB, & understanding the importance of indexes in MongoDB for efficient data querying.	
Day - 3	Index types, working with Object ID, Object ID structure generating & using Object ID in documents.	Master different types of indexes in MongoDB, including single field, compound, text and geospatial indexes.	
Day - 4	Basic query operation find method and query operators, sorting documents. Advanced querying.	Learn to perform basic query operations in MongoDB, including using the find method & query operators for document filtering.	
Day - 5	Aggregation framework Grouping and pipeline stages Map-Reduce for complex aggregations.	Master MongoDB's aggregation framework, including grouping and pipeline stages, for advanced data aggregation operations.	
Day - 6	Database connectivity - MongoDB connect MongoDB with node.js app perform CRUD with MongoDB.	Master MongoDB database connectivity with Node.js applications, including setting up the connection & performing CRUD operations to interact with MongoDB data base efficiently.	

of Month

WEEKLY REPORT

WEEK - 4 (From Dt..... to Dt.....)

Objective of the Activity Done:

Detailed Report:

INTRODUCTION To MONGODB

- Learn about MongoDB and its advantages as a NoSQL database for flexible and scalable data storage.
- Understand the basics of setting up a MongoDB development environment.

Building MongoDB schema and Models with Mongoose

- Gain proficiency in defining schemas and models for MongoDB using Mongoose, a popular Node.js library.
- Learn how to structure data effectively for MongoDB storage.

Using Mongoose to perform CRUD Operations

- Master CRUD (Create, Read, Update, Delete) operations in MongoDB using Mongoose.
- Understand how to interact with MongoDB databases to manage data efficiently.

Implementing Authentication & Authorization with MONGODB

- Learn to implement user authentication and authorization using MongoDB.

ACTIVITY LOG FOR THE FIFTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day - 1	Connecting Frontend & Backend Making API calls from the frontend to interact with the backend using Fetch API.	Master connecting frontend & backend systems by making API calls from the frontend to the backend using the Fetch API.	
Day - 2	Node.js Advanced Concepts and Authentication understanding the Node.js event loop & asynchronous programming Using callbacks, promises, & async/await in Node.js.	Node.js Advanced concepts & Authentication understanding the Node.js event loop & asynchronous programming using callbacks, promises, & async/await in Node.js.	
Day - 3	Handling errors & debugging Node.js applications Implementing security best practices in Node.js	Handling error and debugging Node.js applications Implementing security best practices in Node.js	
Day - 4	Error handling & validation Error handling in express custom error handling middleware Handling different types of errors Input validation using libraries like Joi for data validation.	Master error handling and validation in express.js. Learn input validation techniques using libraries like Joi for ensuring data integrity & security.	
Day - 5	Performance Optimization Techniques to optimize the performance of MERN applications, including code splitting & lazy loading.	Master performance optimization techniques for MERN applications, including code splitting & lazy loading, to improve speed & user experience.	
Day - 6	Deployment & Hosting Introduction to CI & CD pipelines Deploying Node.js applications with server configurations.	Master deployment & hosting of Node.js applications, including setting up CI/CD pipelines for automated deployment.	af. PhutV

WEEKLY REPORT

WEEK - 5 (From Dt..... to Dt.....)

Objective of the Activity Done:

Detailed Report:

CONNECTING FRONTEND AND BACKEND:

- Learn to integrate frontend and backend systems by making API calls using the Fetch API.
- Understand the process of sending and receiving data between the two layers for seamless interaction in web applications.

Node.js Advanced Concepts & Authentication:

- Gain a deep understanding of the Node.js event loop and asynchronous programming.
- Master the use of callbacks, promises, and async/await in Node.js for efficient handling of asynchronous operations.
- Learn to implement authentication mechanisms in Node.js applications for secure user authentication.

Handling Errors and Debugging Node.js Applications:

- Develop skills in identifying and resolving errors in Node.js applications.
- Implement best practices for debugging to improve the performance & reliability of Node.js applications.

ACTIVITY LOG FOR THE SIXTH WEEK

Day & Date	Brief description of the daily activity	Learning Outcome	Person In-Charge Signature
Day - 1	Setting up a React development environment. Setting up the Development Environment.	Master the process of Setting up a React development environment, including configuration tools & dependencies.	
Day - 2	Setting up an Express application Routing & middleware concept. Setting up a MongoDB development environment.	Setting up an express application involves configuration routes and middleware, essential for building robust web servers.	
Day - 3	React components, Props & Conditional Rendering. core modules, commonJS modules and npm packages.	Learn to create and manage React components. Use props for data transfer, & implement conditional rendering for dynamic user interfaces.	
Day - 4	Handling HTTP requests & Responses. Creating routes & parsing request data and sending responses. Using Mongoose to perform CRUD operations in MongoDB.	Master handling HTTP requests & responses in express.js, including creating routes & parsing request data for sending responses.	
Day - 5	Basic query Operations find method and query operators.	Learn to perform basic query operations in MongoDB, including using the find method & query operators for document filtering.	
Day - 6	Aggregation framework Grouping and pipeline stages.	Master the MongoDB aggregation framework, including grouping and pipeline stages, for advanced data aggregation operations.	

Al. Murtuza

WEEKLY REPORT

WEEK - 6 (From Dt..... to Dt.....)

Objective of the Activity Done:

Detailed Report:

Setting up a React development Environment:

- Learn to configure tools and dependencies for a stable React development environment.

Setting up an express Application with Routing & Middleware :-

- Configure routes and middleware in express for building robust web servers.
- Understand the essential concepts of routing and middleware for effective request handling.

Setting up a MongoDB Development Environment:-

- Set up a MongoDB development environment for efficient database management & interaction.
- Learn the basics of MongoDB, including schema design and data modeling.

React components, Props & conditional Rendering:

- Master React component creation & Management.
- Understand how to use props for data transfer and implement conditional rendering for dynamic user interfaces.

Evaluation by the Supervisor of the Intern Organization

Student Name: A. Pavan Kumar

Registration No: T2112880
-5092

Term of Internship: 6 Weeks

Date of Evaluation:

Organization Name & Address: SMART BRIDGE- DATA ANALYTICS
VIRTUAL INTERNSHIP

Please rate the student's performance in the following areas:

Please note that your evaluation shall be done independent of the Student's self-evaluation

Rating Scale: 1 is lowest and 5 is highest rank

1	Oral communication	1	2	3	4	5
2	Written communication	1	2	3	4	5
3	Proactiveness	1	2	3	4	5
4	Interaction ability with community	1	2	3	4	5
5	Positive Attitude	1	2	3	4	5
6	Self-confidence	1	2	3	4	5
7	Ability to learn	1	2	3	4	5
8	Work Plan and organization	1	2	3	4	5
9	Professionalism	1	2	3	4	5
10	Creativity	1	2	3	4	5
11	Quality of work done	1	2	3	4	5
12	Time Management	1	2	3	4	5
13	Understanding the Community	1	2	3	4	5
14	Achievement of Desired Outcomes	1	2	3	4	5
15	OVERALL PERFORMANCE	1	2	3	4	5

Date:

Signature of the Supervisor

Student Self Evaluation of the Short-Term Internship

Student Name: A Pavan Kumar

Registration No: T2112880
-5092

Term of Internship: 6 weeks From: 11092023

To: 20102023

Date of Evaluation:

Organization Name & Address: SMART BRIDGE, DATA ANALYTICS
VIRTUAL INTERNSHIP.

Please rate your performance in the following areas:

Rating Scale:

Letter grade of CGPA calculation to be provided

1	Oral communication	1	2	3	4	5
2	Written communication	1	2	3	4	5
3	Proactiveness	1	2	3	4	5
4	Interaction ability with community	1	2	3	4	5
5	Positive Attitude	1	2	3	4	5
6	Self-confidence	1	2	3	4	5
7	Ability to learn	1	2	3	4	5
8	Work Plan and organization	1	2	3	4	5
9	Professionalism	1	2	3	4	5
10	Creativity	1	2	3	4	5
11	Quality of work done	1	2	3	4	5
12	Time Management	1	2	3	4	5
13	Understanding the Community	1	2	3	4	5
14	Achievement of Desired Outcomes	1	2	3	4	5
15	OVERALL PERFORMANCE	1	2	3	4	5

Date:

A Pavan Kumar
Signature of the Student

ASSESSMENT STATEMENT

Name Of the Student: A.Pavan Kumar
Programme of Study: Bachelor of Science
Year of Study: 2021- 2024
Group: MPCS B
Register No/H.T. No: 721128805092

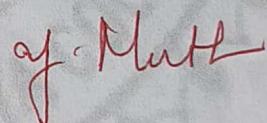
Name of the College: Dr. Lankapalli Bullayya College

University: Andhra University

Sl.No	Evaluation Criterion	Maximum Marks	Marks Awarded
1.	Activity Log	10	7
2.	Internship Evaluation	30	30
3.	Oral Presentation	10	10
	GRAND TOTAL	50	47

Date:

Signature of the Faculty Guide



Signature of the Head of the Department:

Signature of the Principal with Seal

