SCTR's Pune Institute of Computer Technology Dhankawadi, Pune

A.Y. 2023-24

WADL MINI PROJECT REPORT ON

"Discussion Forum"

Submitted By

Rutuja Vanyalkar – 33389
Sayali Walkoli-33393

Under the guidance of
GUIDE NAME- Mrs.D.P.Salapurkar

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR 2023-24

ABSTRACT

The project encompasses the development and implementation of an interactive online platform tailored to facilitate meaningful discussions, knowledge sharing, and collaboration among participants. Key features include user-friendly interface design, robust moderation tools, personalized user profiles, and categorization of discussions to streamline content navigation Central to the project's objectives is the cultivation of a vibrant and inclusive community where diverse perspectives are respected and constructive dialogue is encouraged. Through effective moderation policies and user guidelines, the platform aims to foster a safe and welcoming environment conducive to productive discourse. Furthermore, the project incorporates mechanisms for incentivizing participation and rewarding valuable contributions, thereby incentivizing active engagement and knowledge sharing among users. Features such as gamification elements, badges, and reputation systems are integrated to acknowledge and promote exemplary contributions within the community.

INTRODUCTION

In today's interconnected digital landscape, online discussion forums serve as indispensable platforms for fostering communication, collaboration, and knowledge exchange across diverse communities and interest groups. These forums provide spaces where individuals from different backgrounds, geographies, and perspectives can come together to engage in meaningful discourse, share insights, and collectively address complex issues.Recognizing the pivotal role of online discussion forums in facilitating dialogue and community building, the following project aims to develop and implement an innovative discussion forum platform tailored to enhance online discourse within a specific community or interest group. The project endeavors to address the evolving needs and challenges associated with online communication by integrating user-centric design principles, robust moderation mechanisms, and incentives for active participation.

This introduction provides an overview of the discussion forum project's objectives, highlighting its commitment to fostering a vibrant and inclusive community where diverse voices are valued and respected. Through the implementation of advanced features and best practices, the project seeks to create a conducive environment for constructive dialogue, collaborative problem-solving, and knowledge sharing.

LITERATURE SURVEY

Aspect	Discussion Forums	IT Infrastructure
Purpose	Social media platforms for discussion and community engagement, covering diverse topics.	Management body for approving or rejecting change requests, ensuring proper planning and risk assessment.
Example	Flyertalk.com	Change Advisory Board (CAB)
Date	2015	2013
Authors	Krish Krishnan, Shawn P. Rogers	David Watson, Andrew Jones
Description	Discussion forums evolved from early Internet communities to cover various topics, offering valuable social data.	CAB assesses change requests, evaluates risks, reviews failed changes, and facilitates shared learning from the process.
Importance	Valuable for social analytics, providing insights into trends, brand awareness, and sentiment.	Essential for managing changes in IT infrastructure, ensuring stability, and minimizing risks to business operations.
Functionality	Provides a platform for users to discuss topics of interest, share information, and engage with a community.	Approves or rejects change requests, conducts reviews, facilitates discussions, and ensures necessary documentation.
Meeting Frequency	N/A (Not mentioned)	Initially as required, may become regular weekly or monthly meetings as the forensic laboratory grows.
Attendance Requirement	N/A (Not mentioned)	Requestor or their representative must be present; absence can result in automatic rejection of the change request.
Agenda	N/A (Not mentioned)	Discuss upcoming changes, prepare for implementations, review changes already implemented.

Web technologies used:

- Git: A distributed version control system used for tracking changes in codebase and collaboration among developers.
- GitHub/GitLab/Bitbucket: Platforms for hosting Git repositories and facilitating collaboration on code projects.

Frontend development

When building a discussion forum using React, developers have access to a wide range of libraries and tools that can enhance the development process and improve the user experience. Here are some React libraries commonly used for developing discussion forums:

1. **React Router:**
    - React Router is a popular routing library for React applications. It allows developers to define multiple routes in the application, enabling navigation between different pages or components. In a discussion forum, React Router can be used to navigate between different sections such as home, categories, threads, and user profiles.
2. **React-Query:**
    - React-Query is a library for managing, caching, and updating asynchronous data in React applications. It provides hooks for fetching data from APIs and caching the results, which is useful for fetching and displaying discussion threads, comments, and user profiles in a discussion forum.

Backend development

1. **Node.js:**
    - Node.js is a runtime environment that allows developers to run JavaScript on the server-side. It's commonly used with frameworks like Express.js to build scalable and efficient backend APIs for React applications. Node.js is particularly suitable for real-time applications and asynchronous I/O operations, making it a good choice for handling concurrent user interactions in a discussion forum.

- framework that provides a set of features for building web servers and APIs. It's lightweight, unopinionated, and easy to use, making it a popular choice for building RESTful APIs to handle CRUD operations for managing discussion threads, comments, user authentication, and other backend functionalities in a discussion forum.

2. **MongoDB:**
    - MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It's a popular choice for backend data storage in Node.js applications due to its scalability, flexibility, and ease of use. MongoDB's schema-less nature allows developers to store and retrieve data in a format that closely matches the structure of JavaScript objects, making it well-suited for storing discussion threads, comments, user profiles, and other dynamic data in a discussion forum.

Integration

RESTful APIs:

- Use RESTful APIs to define endpoints on the backend that expose data and functionality needed by the frontend. These endpoints can handle CRUD operations for managing discussion threads, comments, user profiles, and authentication.
- Implement routes in the Express.js backend to handle incoming HTTP requests from the frontend. Each route corresponds to a specific API endpoint and performs the necessary operations (e.g., querying the database, processing data, and returning responses).
- Use HTTP methods (GET, POST, PUT, DELETE) to perform corresponding actions on the backend (e.g., GET for fetching data, POST for creating data, PUT for updating data, DELETE for deleting data).

PROGRAM CODE

```
const express = require("express");
const mongoose = require("mongoose");
const authRoutes = require("./routes/Authroutes");
const postRoutes = require("./routes/Postroutes");
const cookieParser = require("cookie-parser");
const dotenv = require("dotenv");
const { requireAuth, checkUser } = require("./middleware/authMiddleware");
const app = express();
const PORT = process.env.PORT || 5000;

dotenv.config({ path: "config.env" });

//middleware

app.use(express.static("public"));
app.use(express.json());
app.use(cookieParser());

const dbURI = process.env.DATABASE;
mongoose
  .connect(dbURI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then((result) => {
    app.listen(PORT);
    console.log("Connected to db");
  })
  .catch((err) => console.log(err));
app.use(authRoutes);
app.use(postRoutes);

if (process.env.NODE_ENV == "production") {
  app.use(express.static("client/build"));
  const path = require("path");
  app.get("*", function (req, res) {
    res.sendFile(path.resolve(__dirname, "client", "build", "index.html"));
  });
}
```

```
import React from 'react';

import ReactDOM from 'react-dom/client';
import { BrowserRouter } from "react-router-dom";
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

CONCLUSION

In conclusion, the development of web-based platforms such as discussion forums and the management of IT infrastructure relies on a combination of frontend and backend technologies to create robust, interactive, and efficient systems. The integration of these technologies enables the creation of dynamic user interfaces, seamless data exchange, and effective management of backend operations

REFERENCES

- Express.js Documentation. (n.d.). Express - Node.js web application framework. Retrieved from https://expressjs.com/

- MongoDB Documentation. (n.d.). MongoDB Manual. Retrieved from https://docs.mongodb.com/manual/