CHITKARA UNIVERSITY

Project Report for Full Stack Engineering Project

LAMA BLOG-BLOGGING SITE

Introduction:

This project presents a full-stack blogging platform built with React for a dynamic and responsive frontend, Node.js for robust backend development, and MySQL for efficient data management. The application integrates these technologies to deliver a seamless and engaging blogging experience, enabling features such as user authentication, post creation, commenting, and social interactions. Designed for scalability and performance, this stack demonstrates how modern web technologies can combine to create a powerful, userfriendly platform for content creators, bloggers, and online communities, supporting tasks like content publishing, user engagement, and community building.

Approximate duration (in hours) to complete the project:

Ongoing – 110 hours

Team Members along with roll numbers:

- Suman Ranjan- 2210990872
- Saksham-2210990761
- Rudraksh-2210990745

Problem Statement

In today's fast-paced digital landscape, bloggers, content creators, and online communities require scalable, interactive platforms to publish and manage content effectively. Traditional blogging platforms often face significant challenges, including high latency in content loading, inefficient data management, and complex user authentication systems, making them difficult to scale for growing user bases. Many existing solutions rely on outdated architectures, leading to increased maintenance costs, performance bottlenecks, and limited support for real-time features like comments and social interactions. Furthermore, ensuring secure user authentication, seamless content retrieval, and engaging user experiences remains a persistent challenge for developers. Without a modern framework that integrates a high-performance frontend, robust backend, and efficient database management, content creators struggle to deliver dynamic, user-friendly blogging platforms for applications such as personal blogs, professional content hubs, and community-driven websites. This project aims to address these limitations by developing a full-stack blogging platform that combines React for a responsive frontend, Node.js for a scalable backend, and MySQL for efficient data storage. This solution will streamline the process of building a secure, interactive, and high-performance blogging platform for content creators and their audiences.

Objective

The objective of this project is to develop a scalable, full-stack blogging platform that leverages React, Node.js, and MySQL to deliver interactive, secure, and high-performance content creation and management experiences. This project aims to eliminate complexities in web development by utilizing a modular architecture, ensuring efficient data management, and providing low-latency user interactions. By integrating React for a responsive and dynamic frontend, Node.js for robust backend API development, and MySQL for reliable data storage, the project will create a feature-rich platform supporting user authentication, post creation, commenting, and social interactions.

The primary focus is to enhance user engagement, simplify content publishing, and provide an optimized experience for applications such as personal blogs, professional content platforms, and community-driven websites. Additionally, the project will emphasize scalability, cost-effectiveness, and ease of deployment, ensuring bloggers and developers can seamlessly build and manage dynamic blogging platforms without technical complexity

Scope of the Project

This project focuses on developing a scalable, full-stack blogging platform using React, Node.js, and MySQL to provide interactive, secure, and highperformance content creation and management experiences. The scope covers both technical and functional aspects, ensuring a robust and efficient blogging application.

From a technical perspective, the project will implement React for a responsive and dynamic frontend, ensuring smooth user experiences with optimized rendering and fast navigation. Node.js will be used to build a robust backend with RESTful APIs, enabling efficient handling of user requests and content management. MySQL will manage data storage, providing a reliable and scalable database for users, posts, comments, and social interactions. The integration of JWT-based authentication will ensure secure user access, while WebSocket support will enable real-time features like commenting and notifications.

From a functional perspective, the blogging platform will be designed for use cases such as personal blogs, professional content platforms, and communitydriven websites. The project will focus on enhancing user engagement through features like post creation, commenting, likes, and sharing, while ensuring ease of use for content creators and readers. Additionally, it will provide a modular, flexible architecture, allowing future enhancements such as rich media support, third-party integrations, and advanced content filtering or search capabilities

Benefits

- **Scalability and Performance: Utilizes React and Node.js to build a highperformance, modular application that scales effortlessly with growing user bases and content volumes.**
- **Real-Time Interactions: Leverages WebSocket integration for instant updates in comments and social features, ensuring dynamic and engaging user experiences.**
- **Secure Authentication and Access Control: JWT-based authentication with bcrypt ensures robust security for user accounts, with role-based permissions and session management.**
- **Reduced Development Complexity: Combines open-source technologies like React, Node.js, and MySQL to streamline development, making it easier for developers to build and maintain blogging platforms.**
- **Cost-Effective and Efficient: Open-source stack minimizes operational costs, while MySQL's efficient data management reduces resource demands for content storage and retrieval.**
- **Enhanced User Engagement: Provides interactive features like commenting, likes, and sharing, personalizing user experiences and fostering community interaction.**
- **Seamless Content Management: Enables easy creation, editing, and deletion of posts with rich text editing and image upload capabilities, improving usability for content creators.**
- **Optimized for High-Traffic Applications: React's fast rendering and Node.js's efficient API handling ensure quick content delivery and SEO optimization, making the platform suitable for large-scale blogging sites.**

Tech Stack

Frontend Development

- **React** – A JavaScript library for building a responsive, component-based frontend with dynamic user interfaces and fast rendering.
- **Tailwind CSS** – A utility-first CSS framework for designing a modern, responsive, and customizable user interface.

Backend & Database

- **Node.js with Express** – A JavaScript runtime and framework for building scalable RESTful APIs and handling server-side logic efficiently.
- **MySQL** – A relational database management system for structured data storage, enabling efficient management of users, posts, comments, and social interactions.

Authentication & Security

- **JWT (JSON Web Tokens)** – A secure authentication mechanism for user login, session management, and protecting routes with token-based authorization.
- **bcrypt** – A password-hashing library for securely storing user credentials and ensuring robust security.

Real-Time Features

- **WebSockets** – Supports real-time updates for interactive features like live commenting and notifications within the application.

Cloud & Deployment

- **Vercel** – A cloud platform for deploying the React frontend with automatic scaling, global CDN support, and seamless integration with Git.
- **Heroku / AWS** – Hosting options for the Node.js backend and MySQL database, ensuring scalability and reliable performance.

DevOps & Version Control

- **Git & GitHub** – For version control, team collaboration, and managing project repositories.
- **CI/CD Pipelines** – Automates testing, builds, and deployments

Conclusion

This project presents a robust full-stack blogging platform built using React, Node.js, and MySQL to deliver interactive, secure, and scalable content creation and management experiences. By leveraging a modular architecture, it simplifies the complexities of web development, ensuring high performance, scalability, and cost efficiency. The integration of React for a dynamic frontend, Node.js for a powerful backend, and MySQL for efficient data management enables seamless and engaging user experiences, making blogging accessible for content creators and communities. The project not only enhances user engagement through features like real-time commenting, secure authentication, and social interactions but also provides a future-proof, modular, and flexible solution for diverse applications such as personal blogs, professional content platforms, and community-driven websites.

Ultimately, this project serves as a powerful framework for building scalable blogging applications, offering security, efficiency, and interactivity while paving the way for future advancements in web-based content management and user engagement.

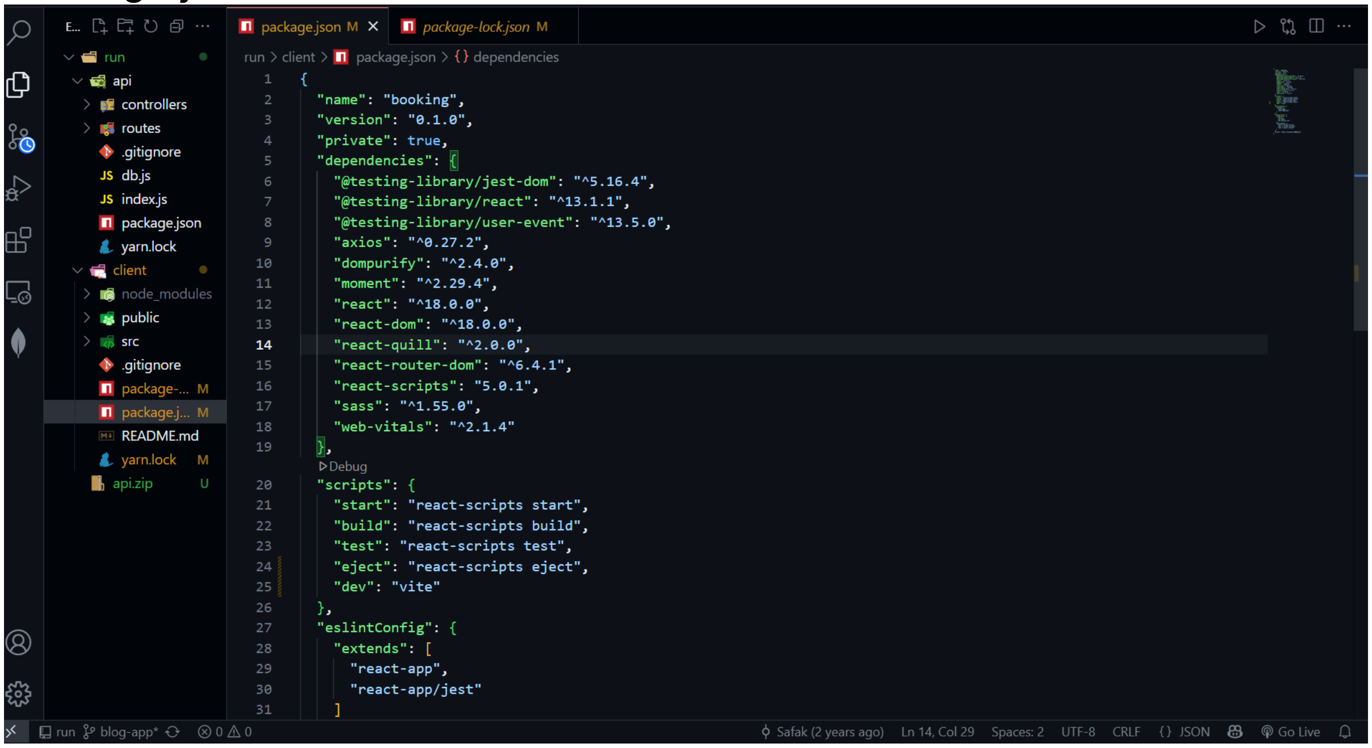
Expected Outcomes

The project will result in a fully functional, full-stack blogging platform that delivers interactive, secure, and scalable content creation and management using React, Node.js, and MySQL. It will provide a modular, high-performance, and cost-effective solution for bloggers, content creators, and developers looking to build engaging online platforms. The blogging site will be capable of dynamic content publishing, real-time social interactions, and secure user management, making it suitable for personal blogs, professional content hubs, and communitydriven websites.

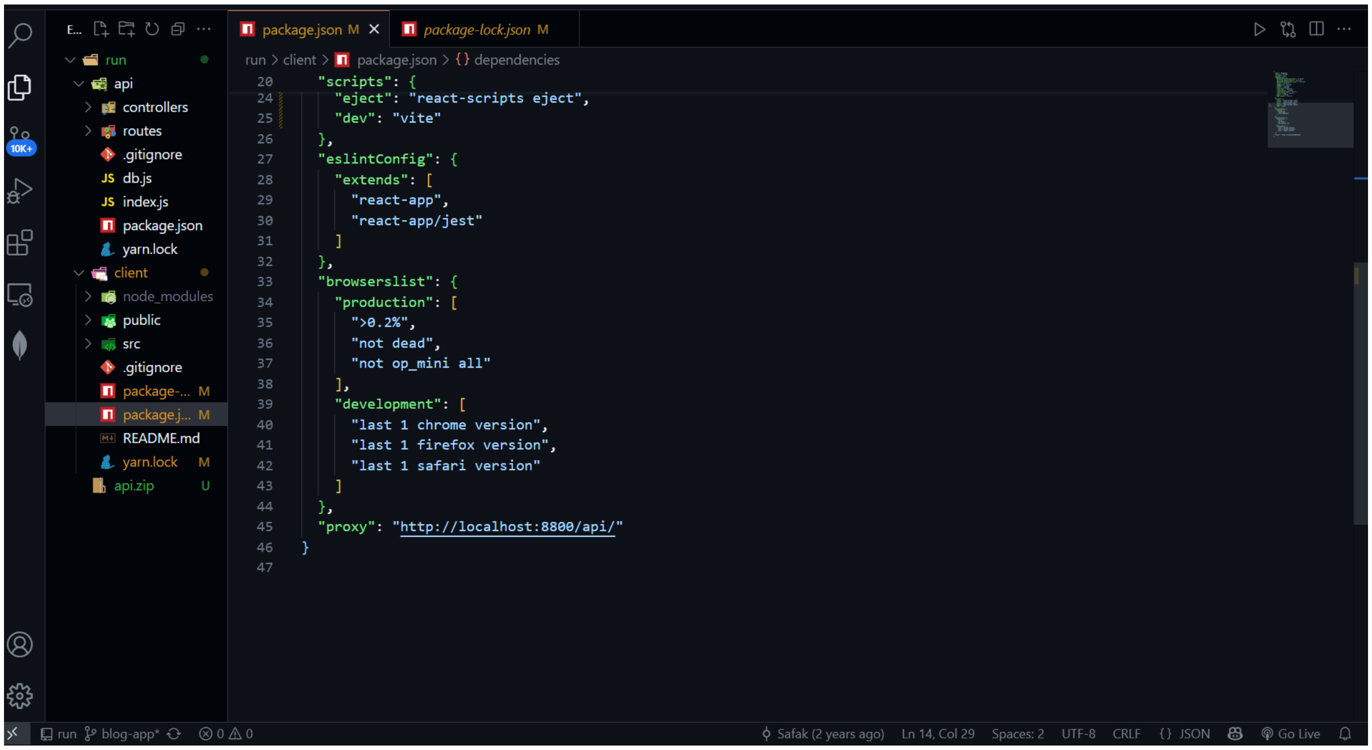
- A fully functional blogging platform with features for post creation, commenting, and social engagement.
- Efficient user account management with secure authentication and authorization.
- Seamless content management through intuitive interfaces for creating, editing, and deleting posts.

Screen Shots of the code

Package.json

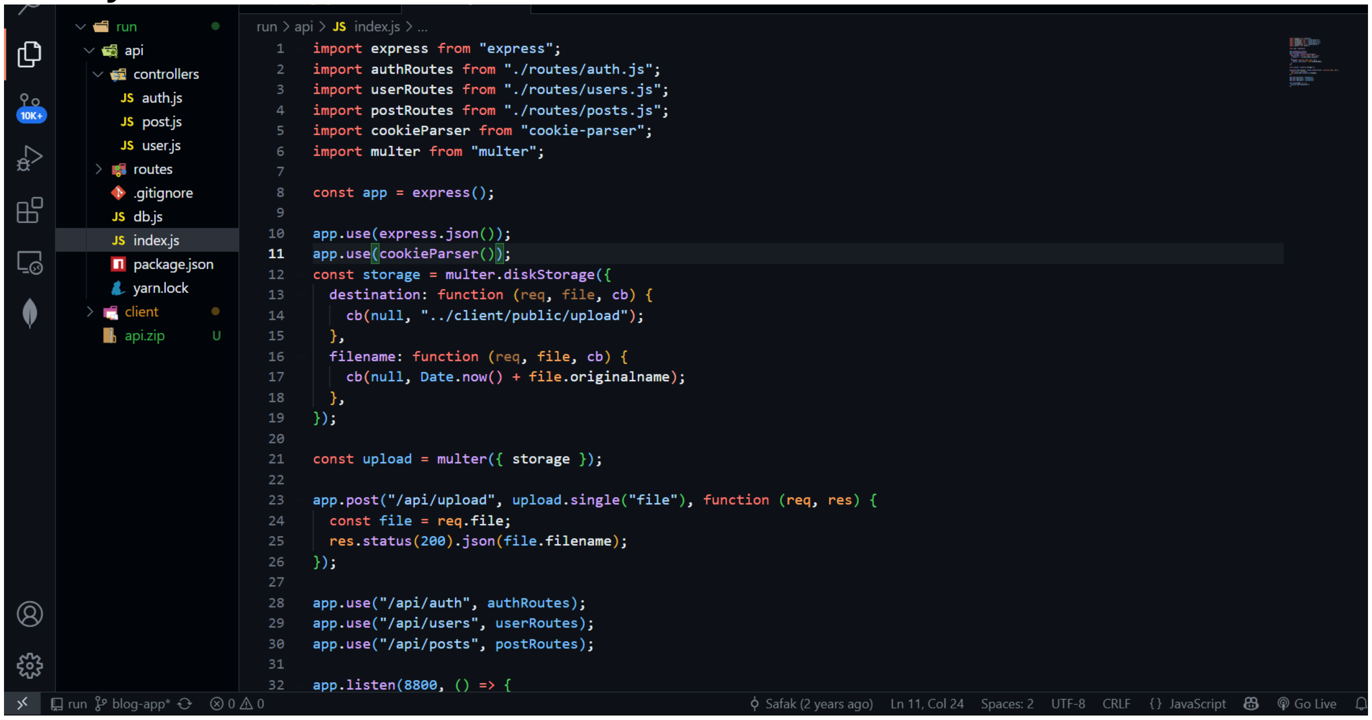


```
{
  "name": "booking",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.1.1",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^0.27.2",
    "dompurify": "^2.4.0",
    "moment": "^2.29.4",
    "react": "^18.0.0",
    "react-dom": "^18.0.0",
    "react-quill": "^2.0.0",
    "react-router-dom": "^6.4.1",
    "react-scripts": "5.0.1",
    "sass": "^1.55.0",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject",
    "dev": "vite"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
```



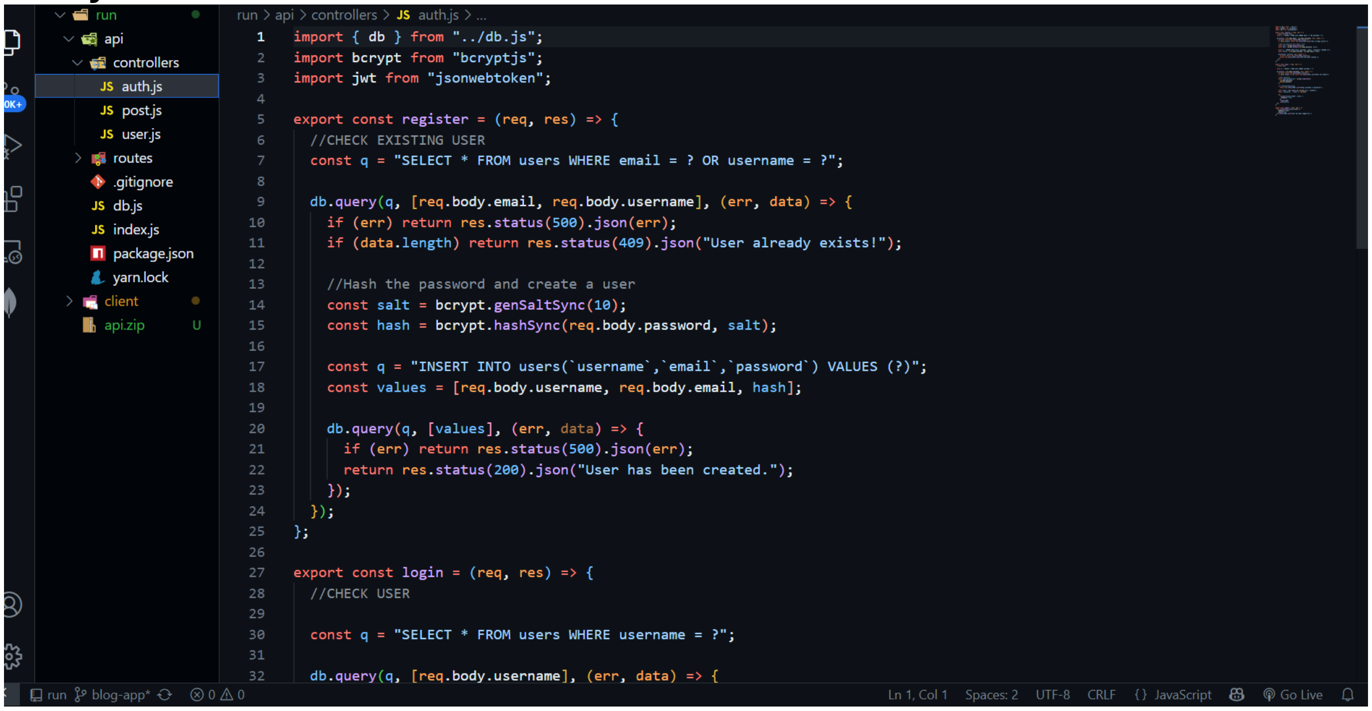
```
  "scripts": {
    "eject": "react-scripts eject",
    "dev": "vite"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "proxy": "http://localhost:8800/api/"
}
```

Index.js

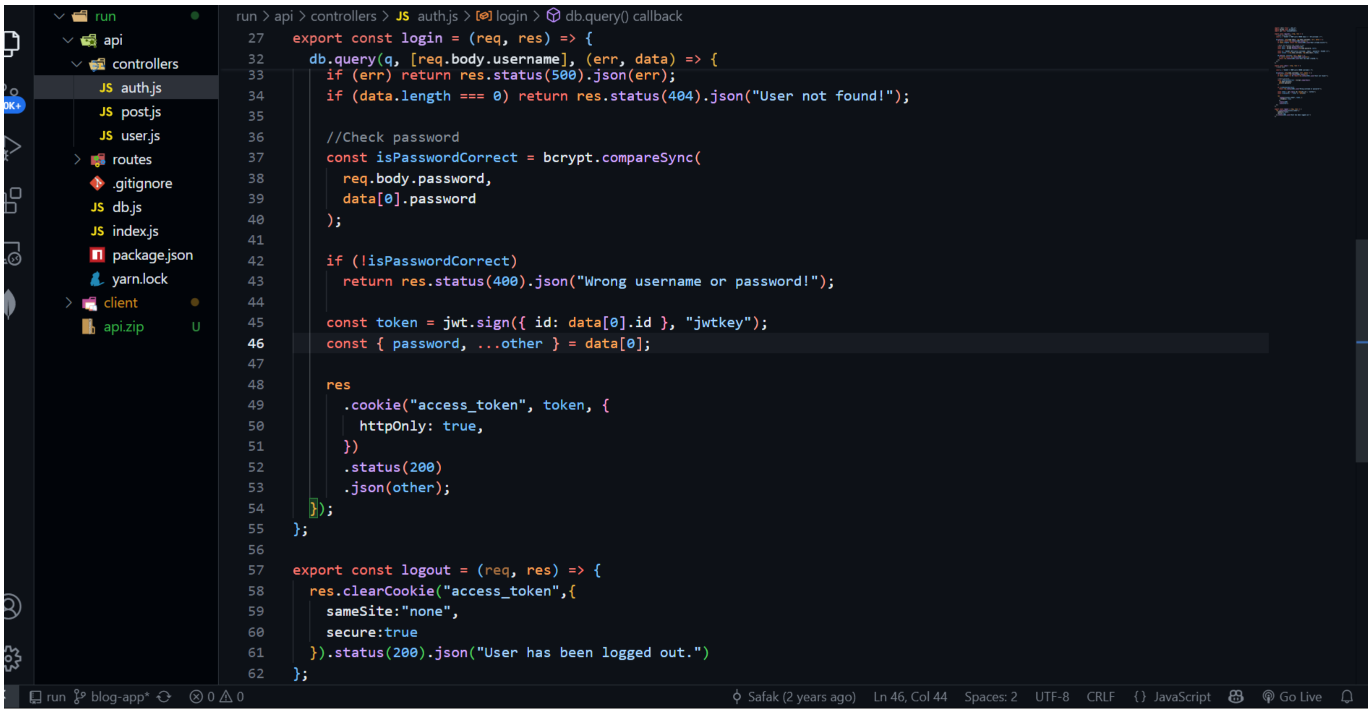

run
api
controllers
auth.js
post.js
user.js
routes
.gitignore
db.js
index.js
package.json
yarn.lock
client
api.zip
run > api > JS index.js > ...
Safak (2 years ago) Ln 11, Col 24 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live

```javascript
import express from "express";
import authRoutes from "./routes/auth.js";
import userRoutes from "./routes/users.js";
import postRoutes from "./routes/posts.js";
import cookieParser from "cookie-parser";
import multer from "multer";

const app = express();

app.use(express.json());
app.use(cookieParser());
const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "../client/public/upload");
  },
  filename: function (req, file, cb) {
    cb(null, Date.now() + file.originalname);
  },
});

const upload = multer({ storage });

app.post("/api/upload", upload.single("file"), function (req, res) {
  const file = req.file;
  res.status(200).json(file.filename);
});

app.use("/api/auth", authRoutes);
app.use("/api/users", userRoutes);
app.use("/api/posts", postRoutes);

app.listen(8800, () => {
```

Auth.js

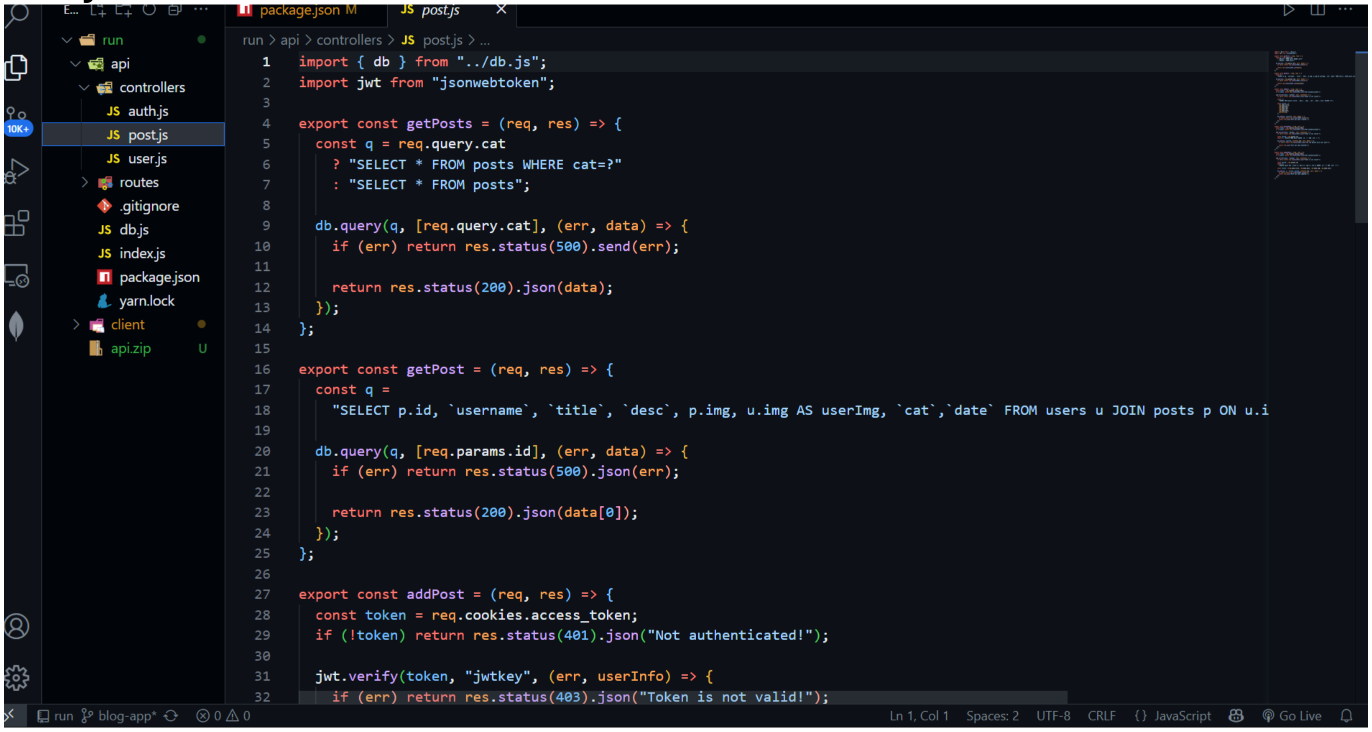

run
api
controllers
auth.js
post.js
user.js
routes
.gitignore
db.js
index.js
package.json
yarn.lock
client
api.zip
run > api > controllers > JS auth.js > ...
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live

```javascript
import { db } from "../db.js";
import bcrypt from "bcryptjs";
import jwt from "jsonwebtoken";

export const register = (req, res) => {
  //CHECK EXISTING USER
  const q = "SELECT * FROM users WHERE email = ? OR username = ?";

  db.query(q, [req.body.email, req.body.username], (err, data) => {
    if (err) return res.status(500).json(err);
    if (data.length) return res.status(409).json("User already exists!");

    //Hash the password and create a user
    const salt = bcrypt.genSaltSync(10);
    const hash = bcrypt.hashSync(req.body.password, salt);

    const q = "INSERT INTO users(`username`,`email`,`password`) VALUES (?)";
    const values = [req.body.username, req.body.email, hash];

    db.query(q, [values], (err, data) => {
      if (err) return res.status(500).json(err);
      return res.status(200).json("User has been created.");
    });
  });
};

export const login = (req, res) => {
  //CHECK USER

  const q = "SELECT * FROM users WHERE username = ?";

  db.query(q, [req.body.username], (err, data) => {
```

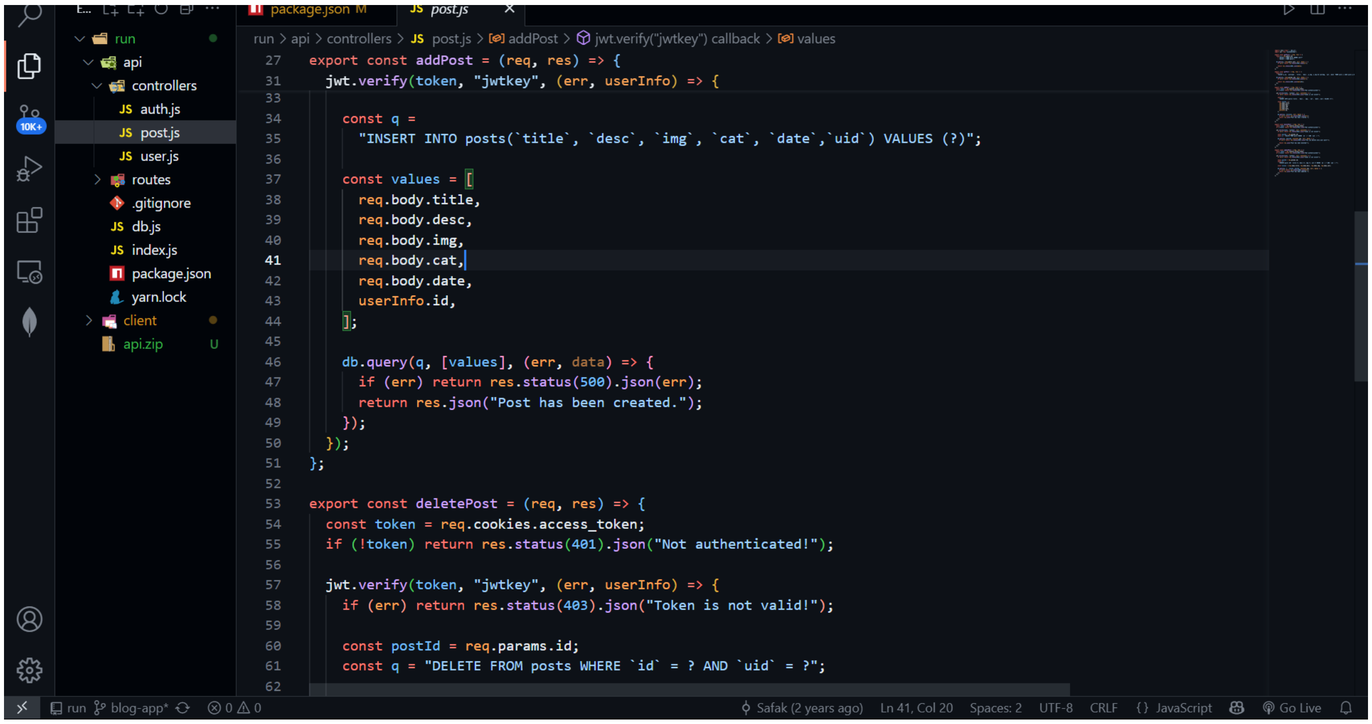


```
export const login = (req, res) => {
  db.query(q, [req.body.username], (err, data) => {
    if (err) return res.status(500).json(err);
    if (data.length === 0) return res.status(404).json("User not found!");

    //Check password
    const isPasswordCorrect = bcrypt.compareSync(
      req.body.password,
      data[0].password
    );

    if (!isPasswordCorrect)
      return res.status(400).json("Wrong username or password!");

    const token = jwt.sign({ id: data[0].id }, "jwtkey");
    const { password, ...other } = data[0];

    res
      .cookie("access_token", token, {
        httpOnly: true,
      })
      .status(200)
      .json(other);
  });
};

export const logout = (req, res) => {
  res.clearCookie("access_token",{
    sameSite:"none",
    secure:true
  }).status(200).json("User has been logged out.")
};
```

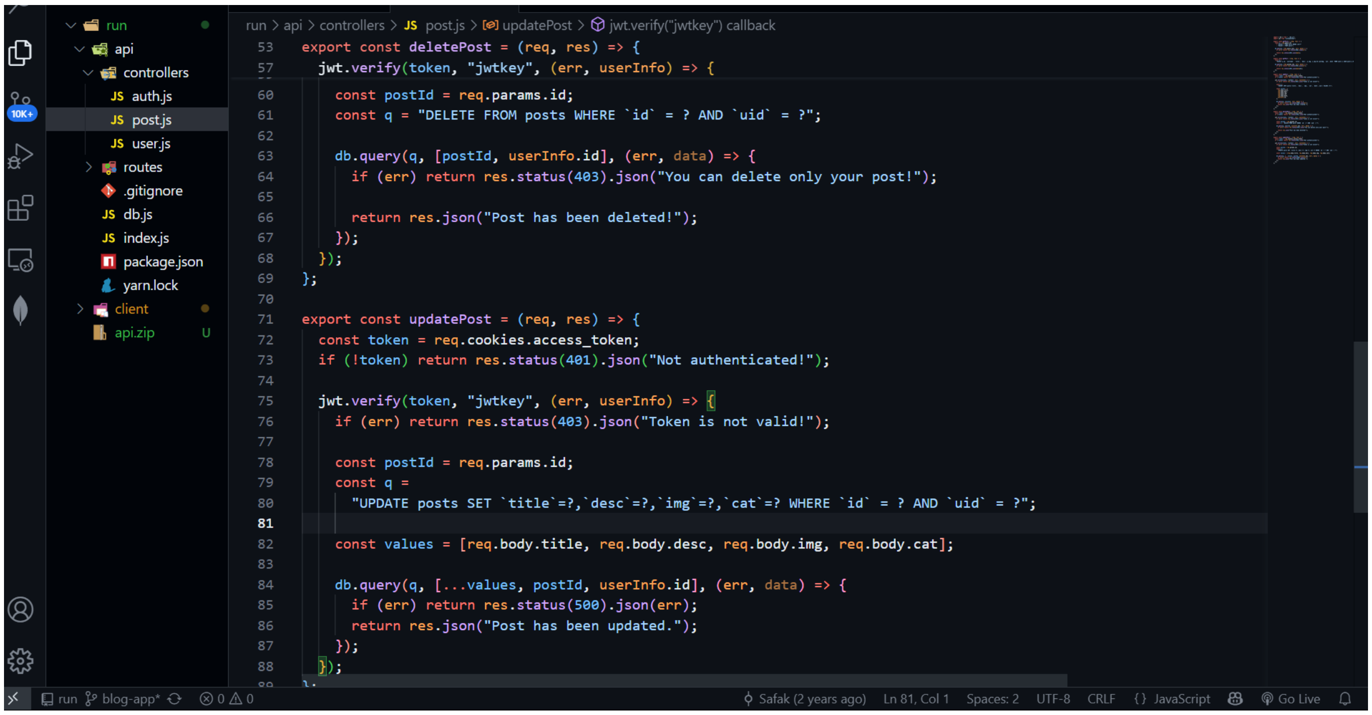
Post.js



```
import { db } from "../db.js";
import jwt from "jsonwebtoken";

export const getPosts = (req, res) => {
  const q = req.query.cat
    ? "SELECT * FROM posts WHERE cat=?"
    : "SELECT * FROM posts";

  db.query(q, [req.query.cat], (err, data) => {
    if (err) return res.status(500).send(err);

    return res.status(200).json(data);
  });
};

export const getPost = (req, res) => {
  const q =
    "SELECT p.id, `username`, `title`, `desc`, p.img, u.img AS userImg, `cat`,`date` FROM users u JOIN posts p ON u.i

  db.query(q, [req.params.id], (err, data) => {
    if (err) return res.status(500).json(err);

    return res.status(200).json(data[0]);
  });
};

export const addPost = (req, res) => {
  const token = req.cookies.access_token;
  if (!token) return res.status(401).json("Not authenticated!");

  jwt.verify(token, "jwtkey", (err, userInfo) => {
    if (err) return res.status(403).json("Token is not valid!");
```

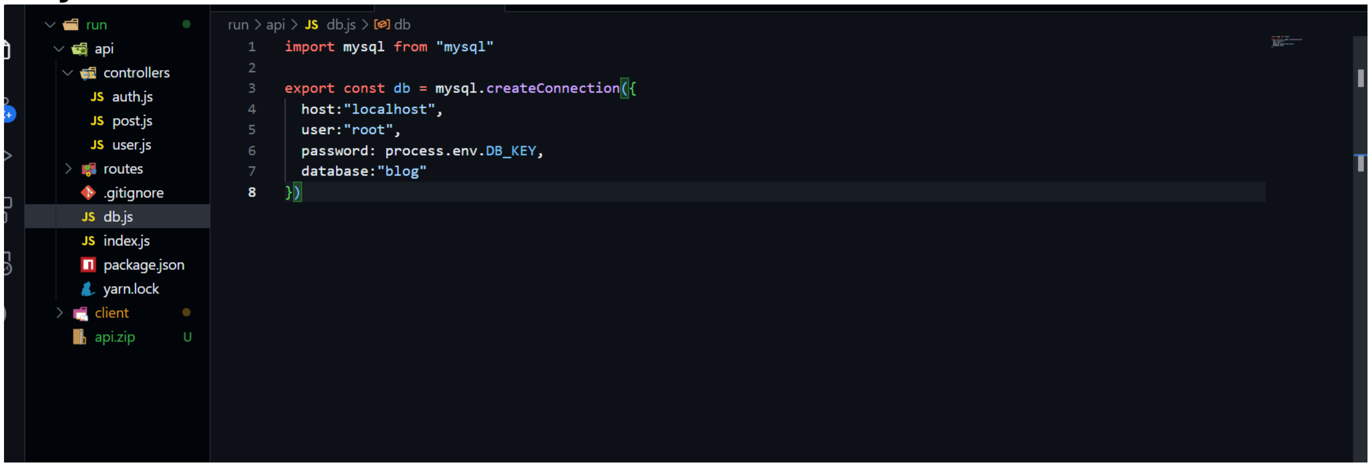

package.json M
post.js
run > api > controllers > post.js > addPost > jwt.verify("jwtkey") callback > values
run
api
controllers
auth.js
post.js
user.js
routes
.gitignore
db.js
index.js
package.json
yarn.lock
client
api.zip
run blog-app* 0 0
Safak (2 years ago) Ln 41, Col 20 Spaces: 2 UTF-8 CRLF JavaScript Go Live

```
export const addPost = (req, res) => {
  jwt.verify(token, "jwtkey", (err, userInfo) => {

    const q =
      "INSERT INTO posts(`title`, `desc`, `img`, `cat`, `date`,`uid`) VALUES (?)";

    const values = [
      req.body.title,
      req.body.desc,
      req.body.img,
      req.body.cat,
      req.body.date,
      userInfo.id,
    ];

    db.query(q, [values], (err, data) => {
      if (err) return res.status(500).json(err);
      return res.json("Post has been created.");
    });
  });
};

export const deletePost = (req, res) => {
  const token = req.cookies.access_token;
  if (!token) return res.status(401).json("Not authenticated!");

  jwt.verify(token, "jwtkey", (err, userInfo) => {
    if (err) return res.status(403).json("Token is not valid!");

    const postId = req.params.id;
    const q = "DELETE FROM posts WHERE `id` = ? AND `uid` = ?";
```


run > api > controllers > post.js > updatePost > jwt.verify("jwtkey") callback
run
api
controllers
auth.js
post.js
user.js
routes
.gitignore
db.js
index.js
package.json
yarn.lock
client
api.zip
run blog-app* 0 0
Safak (2 years ago) Ln 81, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live

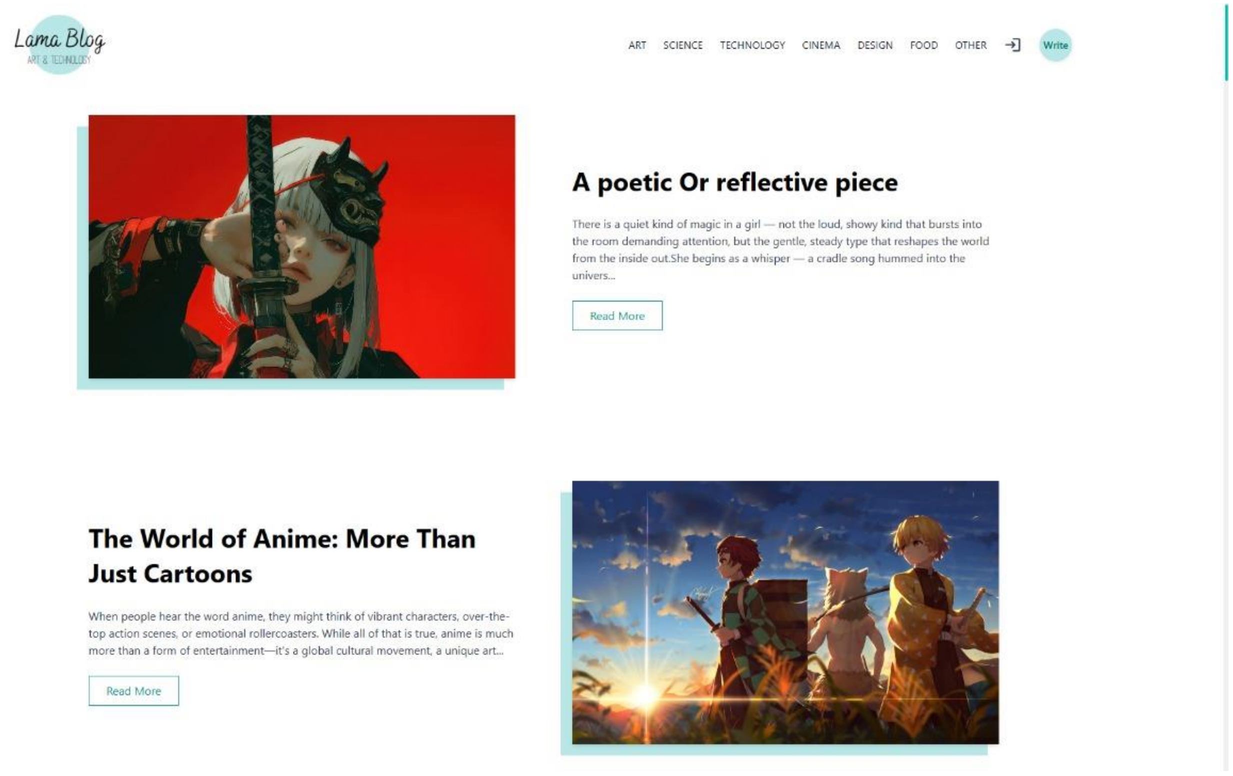
```
export const deletePost = (req, res) => {
  jwt.verify(token, "jwtkey", (err, userInfo) => {

    const postId = req.params.id;
    const q = "DELETE FROM posts WHERE `id` = ? AND `uid` = ?";

    db.query(q, [postId, userInfo.id], (err, data) => {
      if (err) return res.status(403).json("You can delete only your post!");

      return res.json("Post has been deleted!");
    });
  });
};

export const updatePost = (req, res) => {
  const token = req.cookies.access_token;
  if (!token) return res.status(401).json("Not authenticated!");

  jwt.verify(token, "jwtkey", (err, userInfo) => {
    if (err) return res.status(403).json("Token is not valid!");

    const postId = req.params.id;
    const q =
      "UPDATE posts SET `title`=?,`desc`=?,`img`=?,`cat`=? WHERE `id` = ? AND `uid` = ?";

    const values = [req.body.title, req.body.desc, req.body.img, req.body.cat];

    db.query(q, [...values, postId, userInfo.id], (err, data) => {
      if (err) return res.status(500).json(err);
      return res.json("Post has been updated.");
    });
  });
```

Db.js



```
import mysql from "mysql"

export const db = mysql.createConnection({
  host:"localhost",
  user:"root",
  password: process.env.DB_KEY,
  database:"blog"
})
```

Website GUI and Functionalities-



A poetic Or reflective piece

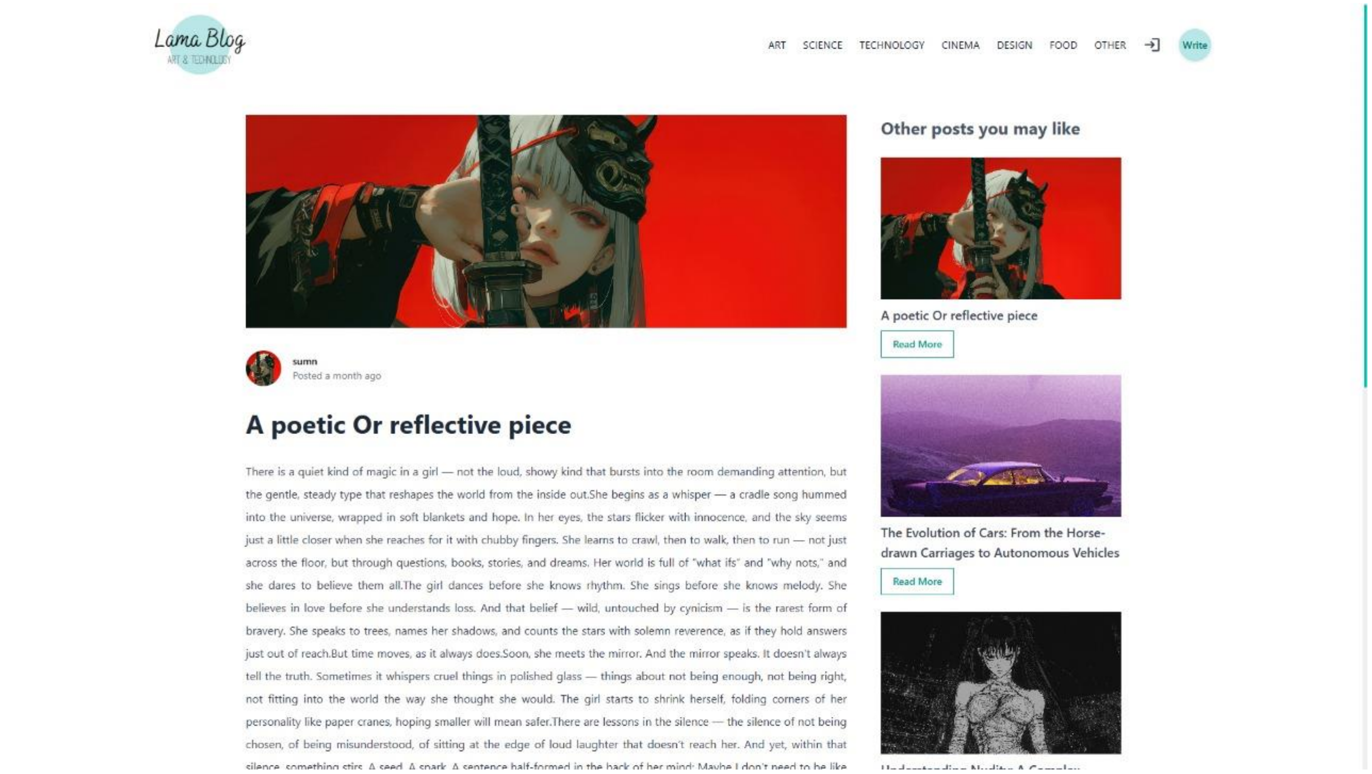
There is a quiet kind of magic in a girl — not the loud, showy kind that bursts into the room demanding attention, but the gentle, steady type that reshapes the world from the inside out.She begins as a whisper — a cradle song hummed into the univers...

Read More

The World of Anime: More Than Just Cartoons

When people hear the word anime, they might think of vibrant characters, over-the-top action scenes, or emotional rollercoasters. While all of that is true, anime is much more than a form of entertainment—it's a global cultural movement, a unique art...

Read More



sumn
Posted a month ago

A poetic Or reflective piece

There is a quiet kind of magic in a girl — not the loud, showy kind that bursts into the room demanding attention, but the gentle, steady type that reshapes the world from the inside out.She begins as a whisper — a cradle song hummed into the universe, wrapped in soft blankets and hope. In her eyes, the stars flicker with innocence, and the sky seems just a little closer when she reaches for it with chubby fingers. She learns to crawl, then to walk, then to run — not just across the floor, but through questions, books, stories, and dreams. Her world is full of "what ifs" and "why nots," and she dares to believe them all.The girl dances before she knows rhythm. She sings before she knows melody. She believes in love before she understands loss. And that belief — wild, untouched by cynicism — is the rarest form of bravery. She speaks to trees, names her shadows, and counts the stars with solemn reverence, as if they hold answers just out of reach.But time moves, as it always does.Soon, she meets the mirror. And the mirror speaks. It doesn't always tell the truth. Sometimes it whispers cruel things in polished glass — things about not being enough, not being right, not fitting into the world the way she thought she would. The girl starts to shrink herself, folding corners of her personality like paper cranes, hoping smaller will mean safer.There are lessons in the silence — the silence of not being chosen, of being misunderstood, of sitting at the edge of loud laughter that doesn't reach her. And yet, within that silence, something stirs. A seed. A spark. A sentence half-formed in the back of her mind: Maybe I don't need to be like

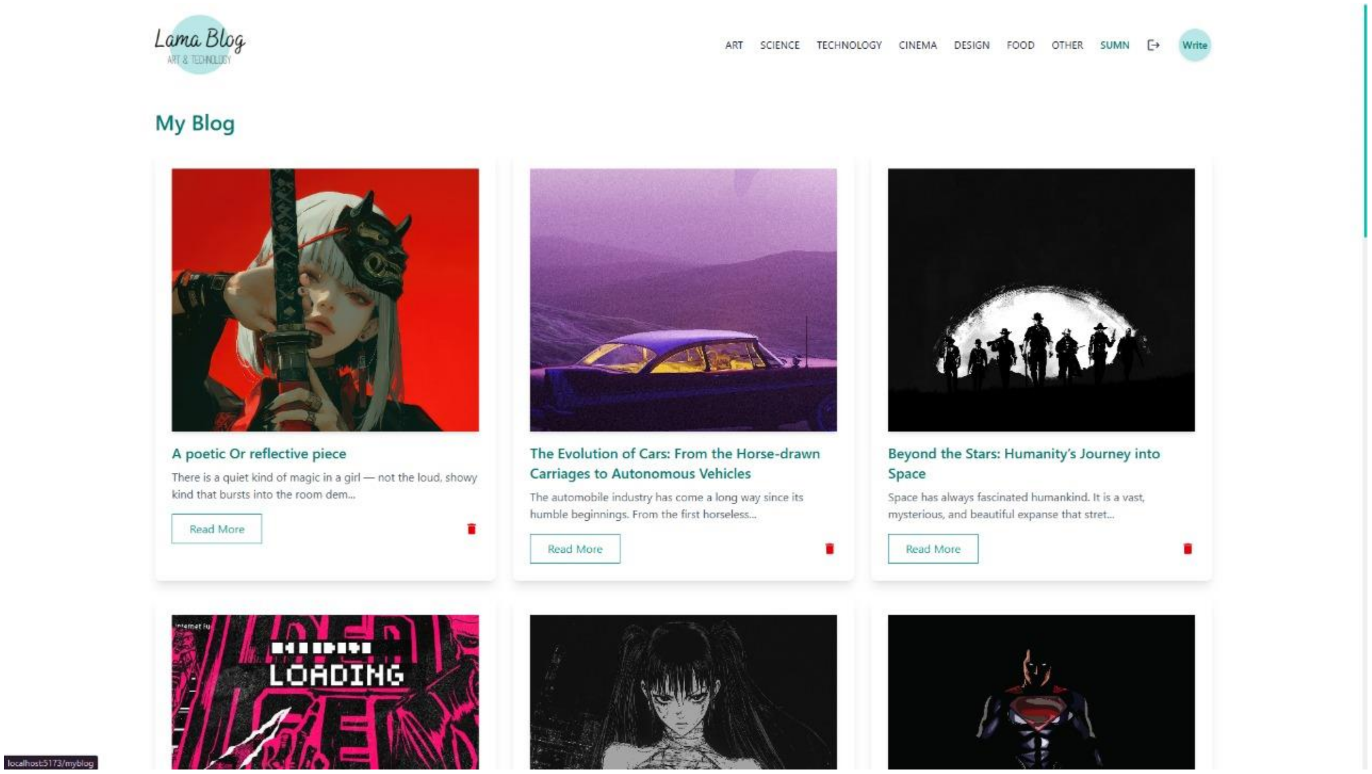
Other posts you may like

A poetic Or reflective piece

Read More

The Evolution of Cars: From the Horse-drawn Carriages to Autonomous Vehicles

Read More



My Blog

A poetic Or reflective piece

There is a quiet kind of magic in a girl — not the loud, showy kind that bursts into the room dem...

Read More

The Evolution of Cars: From the Horse-drawn Carriages to Autonomous Vehicles

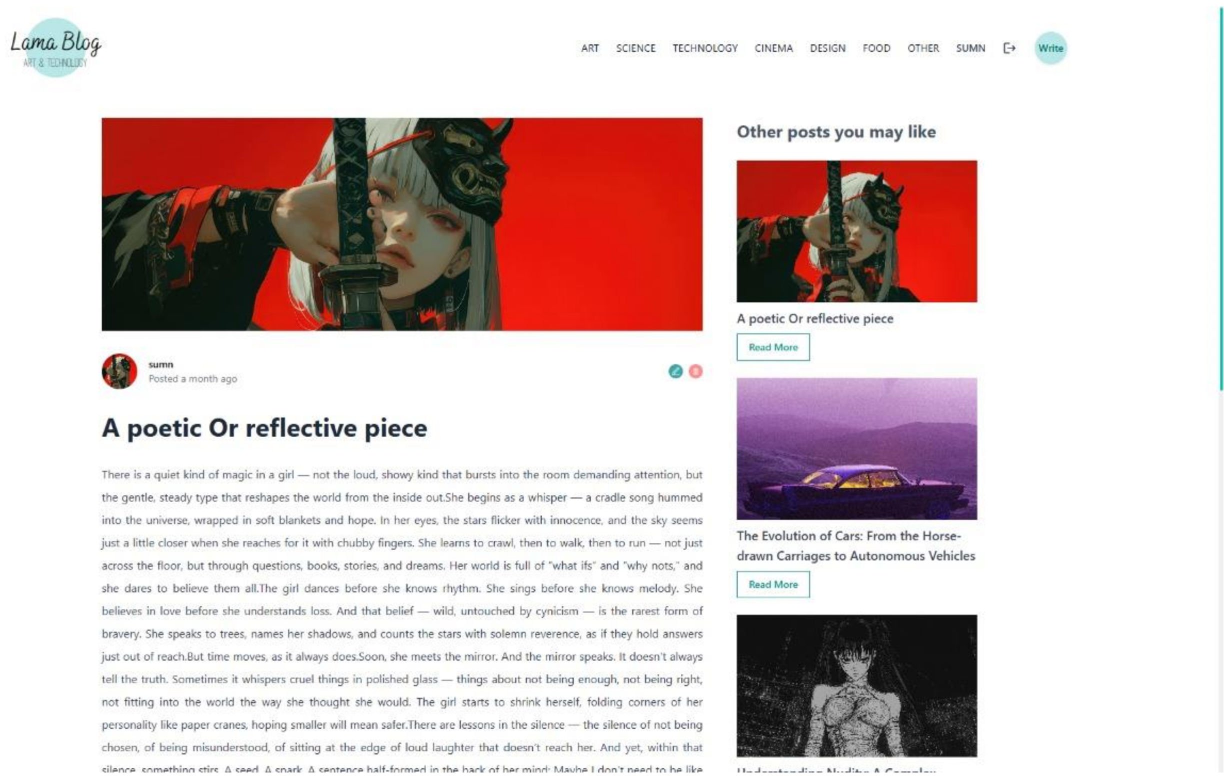
The automobile industry has come a long way since its humble beginnings. From the first horseless...

Read More

Beyond the Stars: Humanity's Journey into Space

Space has always fascinated humankind. It is a vast, mysterious, and beautiful expanse that stret...

Read More



sumn
Posted a month ago

A poetic Or reflective piece

There is a quiet kind of magic in a girl — not the loud, showy kind that bursts into the room demanding attention, but the gentle, steady type that reshapes the world from the inside out.She begins as a whisper — a cradle song hummed into the universe, wrapped in soft blankets and hope. In her eyes, the stars flicker with innocence, and the sky seems just a little closer when she reaches for it with chubby fingers. She learns to crawl, then to walk, then to run — not just across the floor, but through questions, books, stories, and dreams. Her world is full of "what ifs" and "why nots," and she dares to believe them all.The girl dances before she knows rhythm. She sings before she knows melody. She believes in love before she understands loss. And that belief — wild, untouched by cynicism — is the rarest form of bravery. She speaks to trees, names her shadows, and counts the stars with solemn reverence, as if they hold answers just out of reach.But time moves, as it always does.Soon, she meets the mirror. And the mirror speaks. It doesn't always tell the truth. Sometimes it whispers cruel things in polished glass — things about not being enough, not being right, not fitting into the world the way she thought she would. The girl starts to shrink herself, folding corners of her personality like paper cranes, hoping smaller will mean safer.There are lessons in the silence — the silence of not being chosen, of being misunderstood, of sitting at the edge of loud laughter that doesn't reach her. And yet, within that silence, something stirs. A seed. A spark. A sentence half-formed in the back of her mind: Maybe I don't need to be like

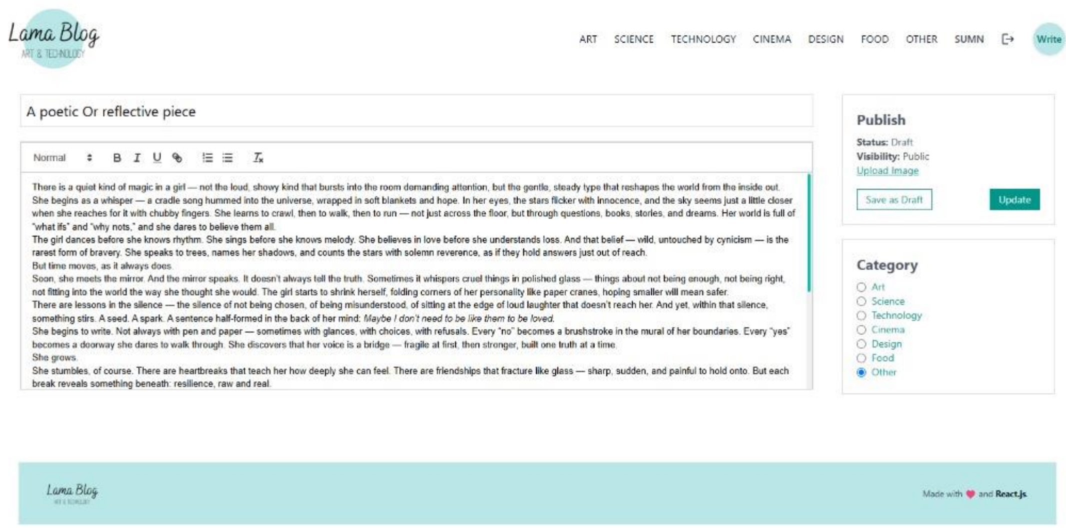
Other posts you may like

A poetic Or reflective piece

Read More

The Evolution of Cars: From the Horse-drawn Carriages to Autonomous Vehicles

Read More



A poetic Or reflective piece

There is a quiet kind of magic in a girl — not the loud, showy kind that bursts into the room demanding attention, but the gentle, steady type that reshapes the world from the inside out.
She begins as a whisper — a cradle song hummed into the universe, wrapped in soft blankets and hope. In her eyes, the stars flicker with innocence, and the sky seems just a little closer when she reaches for it with chubby fingers. She learns to crawl, then to walk, then to run — not just across the floor, but through questions, books, stories, and dreams. Her world is full of "what ifs" and "why nots," and she dares to believe them all.
The girl dances before she knows rhythm. She sings before she knows melody. She believes in love before she understands loss. And that belief — wild, untouched by cynicism — is the rarest form of bravery. She speaks to trees, names her shadows, and counts the stars with solemn reverence, as if they hold answers just out of reach.
But time moves, as it always does.
Soon, she meets the mirror. And the mirror speaks. It doesn't always tell the truth. Sometimes it whispers cruel things in polished glass — things about not being enough, not being right, not fitting into the world the way she thought she would. The girl starts to shrink herself, folding corners of her personality like paper cranes, hoping smaller will mean safer.
There are lessons in the silence — the silence of not being chosen, of being misunderstood, of sitting at the edge of loud laughter that doesn't reach her. And yet, within that silence, something stirs. A seed. A spark. A sentence half-formed in the back of her mind: *Maybe I don't need to be like them to be loved.*
She begins to write. Not always with pen and paper — sometimes with glances, with choices, with refusals. Every "no" becomes a brushstroke in the mural of her boundaries. Every "yes" becomes a doorway she dares to walk through. She discovers that her voice is a bridge — fragile at first, then stronger, built one truth at a time.
She grows.
She stumbles, of course. There are heartbreaks that teach her how deeply she can feel. There are friendships that fracture like glass — sharp, sudden, and painful to hold onto. But each break reveals something beneath: resilience, raw and real.

Publish

Status: Draft
Visibility: Public
Upload Image

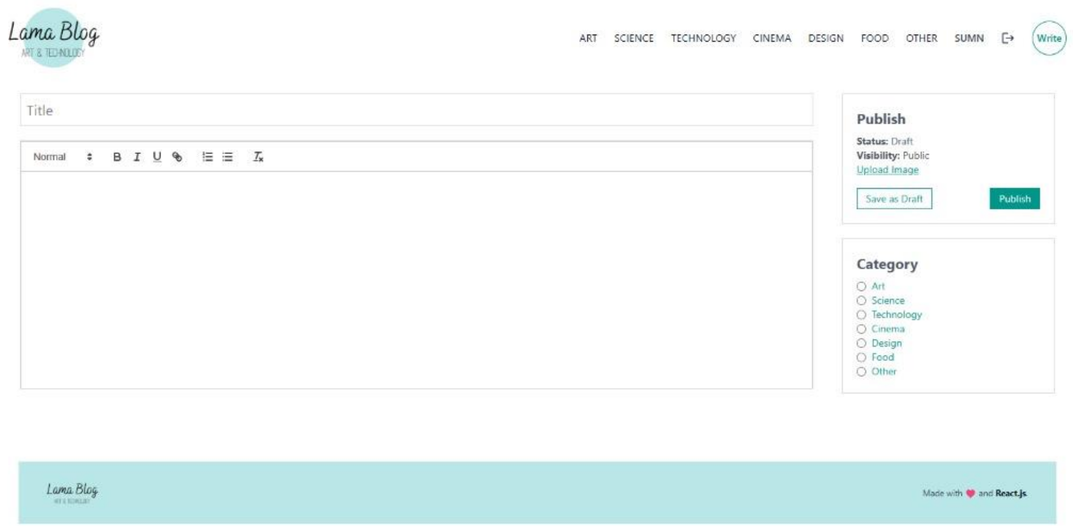
Save as Draft | Update

Category

- ○ Art
- ○ Science
- ○ Technology
- ○ Cinema
- ○ Design
- ○ Food
- ● Other

Lama Blog

Made with ♥ and **React.js**



Resources for References

You Tube: https://www.youtube.com/
Authentication: https://clerk.com/
W3School: https://www.w3schools.com/
Stack Overflow: https://stackoverflow.com/questions