bookKidWorld

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Adarsh Shivam

12115538

INT 222

ADVANCED WEB DEVELOPMENT

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

Month - April Year - 2024

DECLARATION

We hereby declare that the project work entitled “bookKidWorld” is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Divya Thakur, during January to May 2024. All the information furnished in this project report is based on my own intensive work and is genuine.

Name of Student: Adarsh Shivam
Registration Number: 12115538

Adarsh Shivam
(Signature of Student 1)
Date: 28/04/24

CERTIFICATE

This is to certify that the declaration statement made by the student is correct to the best of my knowledge and belief. He have completed this Project under my guidance and supervision. The present work is the result of his original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Signature and Name of the Staff

Designation
School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.
Date : 28/04/24

Acknowledgement

Acknowledgement: This project, "BookKidWorld," harnesses JavaScript, React.js, and Tailwind CSS for frontend development, alongside Node.js for the backend. It offers features such as book discovery, reviews, and purchasing functionalities, catering to children's literature enthusiasts. The platform aims to create an interactive and engaging experience for users exploring the world of children's books.

TABLE OF CONTENTS

CONTENTS **PAGE NO.**

1. INTRODUCTION

Welcome to bookKidWorld, the ultimate destination for young readers and curious minds alike! Our MERN stack project combines the power of JavaScript, React.js, Tailwind CSS, and Node.js to create a dynamic and immersive platform for children's literature enthusiasts.

1.1 Features:

1.1.1 Free Book Section: Explore a vast collection of captivating stories and educational resources available for free. From timeless classics to contemporary tales, there's something for every young reader to enjoy.

1.1.2 Course Section (Paid and login needed): Delve deeper into the world of learning with our premium courses, These courses offer interactive lessons, engaging activities, and exclusive content designed to spark curiosity and ignite a passion for knowledge to the children.

1.1.3 Interactive UI Design: Our user interface is meticulously crafted using React.js and Tailwind CSS, ensuring a seamless and enjoyable browsing experience. Navigate through the site effortlessly, discover new titles, and engage with interactive elements that bring stories to life.

1.1.4 Login and Logout: Sign in to bookKidWorld to unlock personalized features such as customized book recommendations. When you're done exploring, securely logout to protect your account and data.

Whether you're a young reader eager to embark on literary adventures or a parent seeking enriching educational resources, bookKidWorld is your trusted companion on the journey of discovery. Join us and let your imagination soar!

1.2 Benefits:

1.2.1 Educational Enrichment: BookKidWorld offers a plethora of educational resources, including free books and paid courses, designed to enhance children's learning experiences. Through engaging content and interactive activities, children can expand their knowledge and skills in various subjects while having fun.

1.2.2. Convenience and Accessibility: With bookKidWorld's online platform, access to quality children's literature and educational materials is just a few clicks away. Parents, educators, and children can conveniently explore the site from the comfort of their homes or on the go, anytime and anywhere.

1.2.3. Personalized Experience: By logging in to BookKidWorld, users can unlock personalized features tailored to their preferences and learning goals. From customized book recommendations to progress tracking, the platform empowers users to tailor their experience to suit their individual needs.

1.2.4. Engagement and Interaction: The interactive UI design of bookKidWorld fosters engagement and interaction, making the reading and learning experience more enjoyable and immersive for children. From animated story elements to interactive stories, every aspect of the platform is designed to captivate young minds and keep them actively engaged.

1.2.5. Community and Support: bookKidWorld provides a supportive community for parents, educators, and children to connect, share resources, and exchange ideas. Through forums, discussion groups, and social media integration, users can engage with like-minded individuals and foster a sense of belonging within the bookKidWorld community.

1.2.6. Safe and Secure Environment: bookKidWorld prioritizes the safety and privacy of its users, providing a safe and secure online environment for children to explore and learn.

Embark on a rewarding journey of discovery and learning with bookKidWorld, where the benefits of quality children's literature and educational resources await!

2. Profile of the Problem

The "Profile of the Problem" or "Rationale/Scope of the Study" in the context of a website like BookKidWorld refers to the explanation of the problem that the website aims to address, along with the justification for why it's important to solve this problem.

2.1. Problem Statement:

In today's digital age, children are increasingly drawn to screens, spending more time on devices and less time engaging with books and educational materials. This shift in behavior poses significant challenges to children's literacy and cognitive development, leading to concerns about declining reading habits and academic performance.

2.2. Rationale/Scope of the Study:

BookKidWorld aims to tackle this pressing issue by providing a solution that promotes literacy, learning, and engagement among children. By offering a curated selection of free books and paid courses tailored to children's interests and learning needs, the website seeks to:

2.2.1. Encourage Reading: By providing access to a diverse range of captivating stories and educational resources, BookKidWorld aims to instill a love for reading in children and inspire them to explore the world of literature.

2.2.2. Foster Learning: Through interactive courses and engaging activities, BookKidWorld aims to facilitate learning and intellectual development in key subject areas, including language arts, STEM, and social studies.

2.2.3. Promote Digital Literacy: In an increasingly digital world, digital literacy skills are essential for children's success. BookKidWorld provides a safe and interactive online environment where children can develop digital literacy skills while exploring educational content.

2.2.4. Support Parents and Educators: BookKidWorld serves as a valuable resource for parents and educators seeking high-quality children's literature and educational materials. By offering personalized recommendations and progress tracking features, the website empowers parents and educators to support children's learning and monitor their progress effectively.

Overall, BookKidWorld addresses the critical need for accessible, engaging, and educational resources for children, helping to bridge the gap between screen time and quality learning experiences. By leveraging technology to promote literacy and learning, the website aims to empower children to become confident readers, critical thinkers, and lifelong learners.

3. Existing System

3.1. Introduction to Existing System:

- BookKidWorld's existing system is an online platform aimed at providing children with access to a variety of children's books and educational resources. It serves as a digital library where users can discover, read, and engage with books in various formats.

- The existing system allows users to browse through a collection of free books and purchase premium content such as courses and interactive learning materials. It also includes feature for user registration.

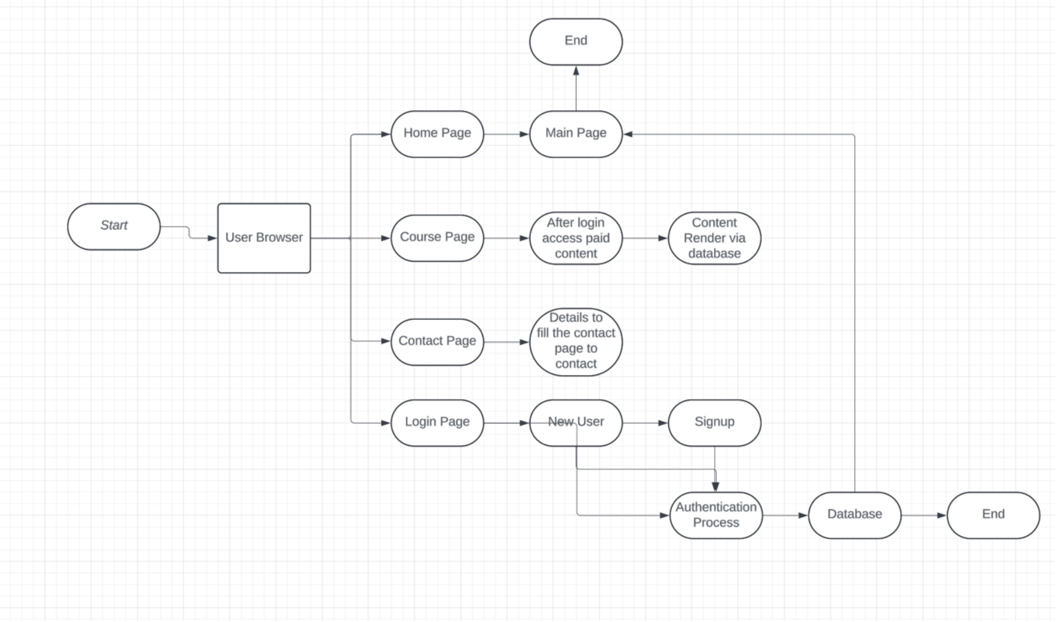
- While the existing system has provided valuable resources for children and parents, there are opportunities to enhance its functionality and user experience to better meet the needs of its audience.

3.2 Existing Software:

- BookKidWorld utilizes a combination of software technologies to support its operations. This includes front-end technologies like React.js for building the user interface and Tailwind CSS for styling.
- On the backend, BookKidWorld relies on Node.js to handle server-side logic and database operations using MongoDB as the database management system.
- Additionally, BookKidWorld may integrate third-party services for features such as payment processing, user authentication, and content delivery.

3.3 DFD for Present System:

- The Data Flow Diagram (DFD) for BookKidWorld's present system illustrates the flow of data within the platform. It shows how data moves from user interactions (such as searching for books or purchasing courses) to backend processes (such as fetching data from the database or processing payments) and back to the user interface.
- The DFD may include components such as user registration, book catalog management, content delivery, and payment processing, interconnected through data flows and processes.



3.4 What’s New in the System to be Developed:

- In the system to be developed, BookKidWorld aims to introduce several new features and enhancements to improve the user experience and expand its capabilities.
- This includes features such as personalized book recommendations based on user preferences and reading history, enhanced course offerings with interactive activities and assessments, and improved performance and scalability to accommodate growing user demand.
- Additionally, the new system may incorporate features for community engagement, such as discussion forums, user-generated content, and social sharing capabilities, to foster a sense of belonging and collaboration among users.

By addressing the existing system's strengths and weaknesses and outlining the new features and improvements to be implemented, BookKidWorld can lay the groundwork for a more robust and engaging platform that better serves its audience of young readers and learners.

4. Problem Analysis

4.1. Problem Analysis:

- In this section, you'll conduct a detailed analysis of the problem that BookStoreKid aims to address. Identify the challenges, pain points, and needs of your target audience, such as children, parents, and educators, in accessing quality children's literature and educational resources.
- Gather feedback from potential users, stakeholders, and market research to understand the demand for a platform like BookStoreKid. Identify any existing solutions or competitors in the market and analyze their strengths and weaknesses.

- Use techniques such as SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) to assess the internal and external factors influencing the success of BookStoreKid.

4.2. Product Definition:

- Define the scope, objectives, and features of BookStoreKid based on the problem analysis. Clearly articulate the value proposition and unique selling points of the platform.
- Identify the target audience and user personas to tailor the product to their needs and preferences. Define user stories and use cases to guide the development process.
- Specify the key functionalities of BookStoreKid, such as browsing and searching for books, accessing free and paid content, user registration and authentication, personalized recommendations, and interactive learning activities.

4.3. Feasibility Analysis:

- Evaluate the technical, financial, and operational feasibility of developing and launching BookStoreKid. Assess the availability of resources, including technology, expertise, and funding, required for the project.
- Consider the scalability and sustainability of the platform in the long term. Identify any technical challenges or constraints that may affect the development and deployment of BookStoreKid.
- Conduct a market analysis to assess the demand for children's literature and educational resources, potential revenue streams, and competitive landscape. Determine the viability of BookStoreKid as a business venture.

4.4. Project Plan:

- Develop a comprehensive project plan outlining the timeline, milestones, and deliverables for the development and launch of BookStoreKid. Define the roles and responsibilities of team members and stakeholders involved in the project.
- Break down the project into manageable tasks and prioritize them based on their importance and dependencies. Create a Gantt chart or timeline to visualize the project schedule and track progress.
- Establish communication channels and workflows for collaboration among team members. Define the project management tools and methodologies to be used, such as Agile or Scrum, to ensure efficient and effective project execution.

By systematically analyzing the problem, defining the product, assessing feasibility, and creating a project plan, you can lay a solid foundation for the development and success of BookStoreKid. This process helps align stakeholders, mitigate risks, and ensure that the project meets the needs and expectations of its target audience.

5. Software Requirement Analysis

5.1. Software Requirement Analysis:

- The introduction provides an overview of the software requirement analysis process for BookStoreKid. It outlines the purpose, scope, and objectives of the analysis, emphasizing the importance of understanding and documenting the requirements before proceeding with development.
- This section may also include a brief explanation of the methodology or approach used for gathering and analyzing requirements, such as interviews, surveys, user stories, or use cases.

5.2. General Description:

- The general description provides a high-level overview of BookStoreKid, including its intended purpose, target audience, and key features. It outlines the broader context in which the software will be used and its expected impact on users and stakeholders.
- Describe the overall functionality and scope of BookStoreKid, highlighting its role in providing access to children's literature and educational resources. Discuss any existing systems or processes that BookStoreKid will integrate with or replace.

5.3. Specific Requirements:

5.3.1 Specific requirements detail the functional and non functional requirements of BookStoreKid in more granular detail. These requirements serve as the basis for designing, developing, and testing the software.

5.3.2 Functional Requirements: Define the specific features, functionalities, and interactions that BookStoreKid must support. This includes user actions, system responses, and any constraints or limitations on behavior.

5.3.3 Non-Functional Requirements: Specify the quality attributes and performance characteristics that BookStoreKid must adhere to. This includes requirements related to usability, reliability, performance, security, and scalability.

5.3.4 User Requirements: Capture the needs, preferences, and expectations of BookStoreKid's target users. This includes requirements related to user interface design, accessibility, and user experience.

5.3.5 System Requirements: Detail the technical specifications and constraints that BookStoreKid must meet. This includes requirements related to hardware, software, infrastructure, and integration with external systems.

5.3.6 Regulatory Requirements: Identify any legal or regulatory requirements that BookStoreKid must comply with, such as data protection regulations or industry standards.

5.3.7 Business Requirements: Define the business objectives, constraints, and success criteria for BookStoreKid. This includes requirements related to revenue generation, cost-effectiveness, market penetration, and competitive differentiation.

By conducting a thorough software requirement analysis, you can ensure that BookStoreKid meets the needs and expectations of its users and stakeholders, while also laying the groundwork for successful design, development, and implementation.

6. Design

6.1. System Design:

6.1.1. Overview: System design involves defining the architecture and components of the BookKidWorld website. It outlines how different parts of the system will interact and work together to achieve the desired functionality.

6.1.2. Key Components: Identify the key components of the website, such as the front-end client, back-end server, database, and any external services or APIs.

6.1.3. Architecture: Determine the overall architecture of the system, such as whether it will follow a monolithic, microservices, or serverless architecture. Define how these components will communicate with each other.

6.1.4 Scalability and Performance: Consider scalability and performance requirements to ensure that the system can handle increasing user loads and maintain responsiveness. This may involve strategies such as load balancing, caching, and optimization techniques.

6.1.5 Security: Address security concerns by implementing measures such as encryption, authentication, authorization, and secure coding practices to protect user data and prevent unauthorized access.

6.2. Design Notations:

6.2.1. Unified Modeling Language (UML): Use UML diagrams such as class diagrams, sequence diagrams, and activity diagrams to visually represent the structure, behavior, and interactions of components within the BookKidWorld system.

6.2.2. Entity-Relationship Diagrams (ERDs): Create ERDs to model the relationships between different entities in the database, such as users, books, courses, and transactions.

6.2.3. Flowcharts: Use flowcharts to depict the flow of logic or processes within the system, such as user registration, authentication, browsing for books, and purchasing courses.

6.2.4. Wireframes and Mockups: Develop wireframes and mockups to visualize the user interface and layout of the website, including page designs, navigation flows, and interactive elements.

6.3. Detailed Design:

6.3.1 Front-end Design: Define the layout, navigation, and user interface components of the BookKidWorld website. This includes designing individual pages, user interfaces for browsing books and courses, user authentication forms, and interactive elements.

6.3.2 Back-end Design: Specify the server-side logic and functionality required to support the front-end features. This includes designing APIs, data models, business logic, and integration points with external services or databases.

6.3.3 Database Design: Design the database schema and structure for storing and managing data related to users, books, courses, transactions, and other entities. Define tables, fields, relationships, and constraints to ensure data integrity and efficiency.

By following these design principles and methodologies, you can create a well-structured, scalable, and user-friendly website for BookKidWorld that meets the needs and expectations of its users and stakeholders.

7. Testing

7.1. Functional Testing:

7.1.1. Definition: Functional testing verifies that the software functions as expected and meets the specified requirements. It focuses on testing individual functions or features of the system to ensure they behave correctly.

7.1.2. Postman Usage: In Postman, you can create test suites to validate the functionality of each API endpoint. Write test scripts in JavaScript using the Postman scripting environment to send requests to the server and assert the expected responses.

Example: For BookKidWorld, functional testing with Postman would involve sending requests to endpoints such as user registration, book retrieval, course purchase, and authentication. Test scripts would verify that users can register successfully, retrieve books and courses, make purchases, and log in/out without errors.

7.2. Structural Testing:

7.2.1. Definition: Structural testing, also known as white-box testing, examines the internal structure of the software to ensure it behaves as designed. It includes testing individual components, code paths, and logic flows within the system.

7.2.2. Postman Usage: While Postman primarily focuses on functional testing, you can indirectly perform structural testing by analyzing the responses returned by the server. Inspect the JSON or XML responses to ensure they contain the expected data fields, structures, and values.

Example: In BookKidWorld, structural testing with Postman would involve validating that the server returns the correct data format and structure for different API endpoints. For example, when retrieving a list of books, ensure that each book object contains the expected fields such as title, author, genre, and publication year.

7.3. Levels of Testing:

7.3.1 Unit Testing: Test individual units or components of the system in isolation, such as functions, methods, or classes. Use frameworks like Mocha or Jest to write and execute unit tests for server-side logic and API endpoints.

7.3.2 Integration Testing: Verify that different components of the system work together as expected. Use Postman to test API integrations and interactions between the client and server, ensuring seamless communication and data exchange.

7.3.3 System Testing: Evaluate the system as a whole to validate end-to-end functionality and performance. Conduct comprehensive testing with Postman to simulate user interactions and workflows, covering all aspects of the application from user registration to content browsing and purchasing.

7.4 Testing the Project with Postman:

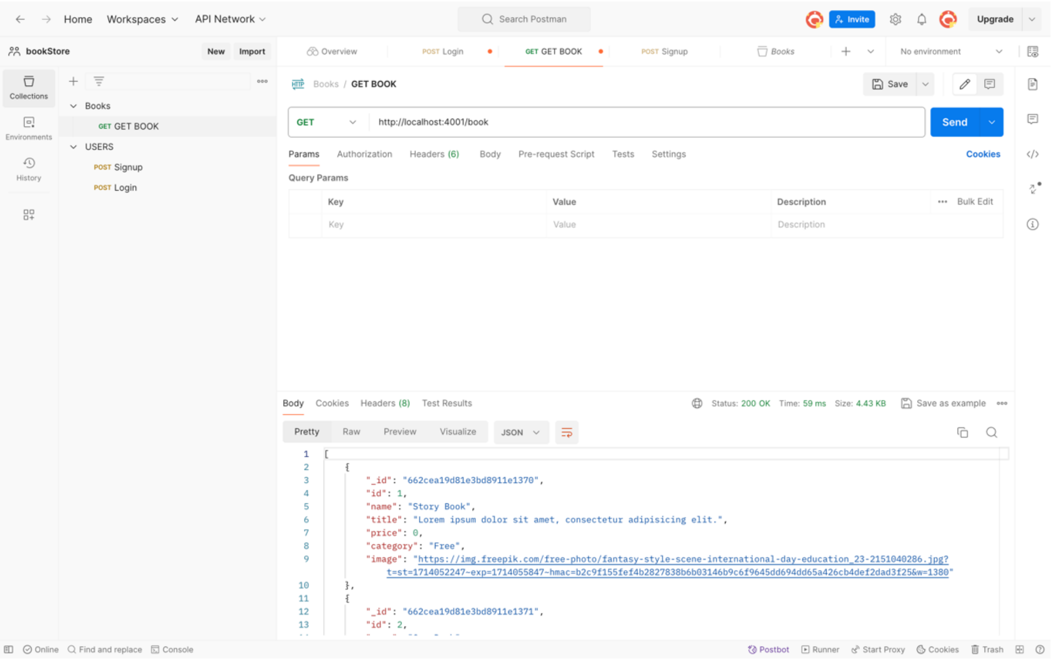
7.4.1. Test Plan: Develop a test plan outlining the objectives, scope, and approach for testing the BookKidWorld project with Postman. Identify the APIs to be tested, test scenarios, and expected outcomes for each scenario.

7.4.2. Test Scripts: Write test scripts in Postman to automate the execution of test cases. Use Postman's scripting capabilities to set up preconditions, send requests, validate responses, and assert expected outcomes.

7.4.3. Test Execution: Execute the test scripts in Postman to run the tests against the backend server. Monitor the test results and debug any failures or errors encountered during testing.

7.4.4. Reporting: Generate test reports in Postman to document the test results, including passed tests, failed tests, and any issues identified. Use the reports to track progress, prioritize fixes, and communicate testing outcomes to stakeholders.

By leveraging Postman for functional testing of the backend API in BookKidWorld, you can ensure the reliability, functionality, and performance of the system, delivering a high-quality product that meets user expectations.



bookStore

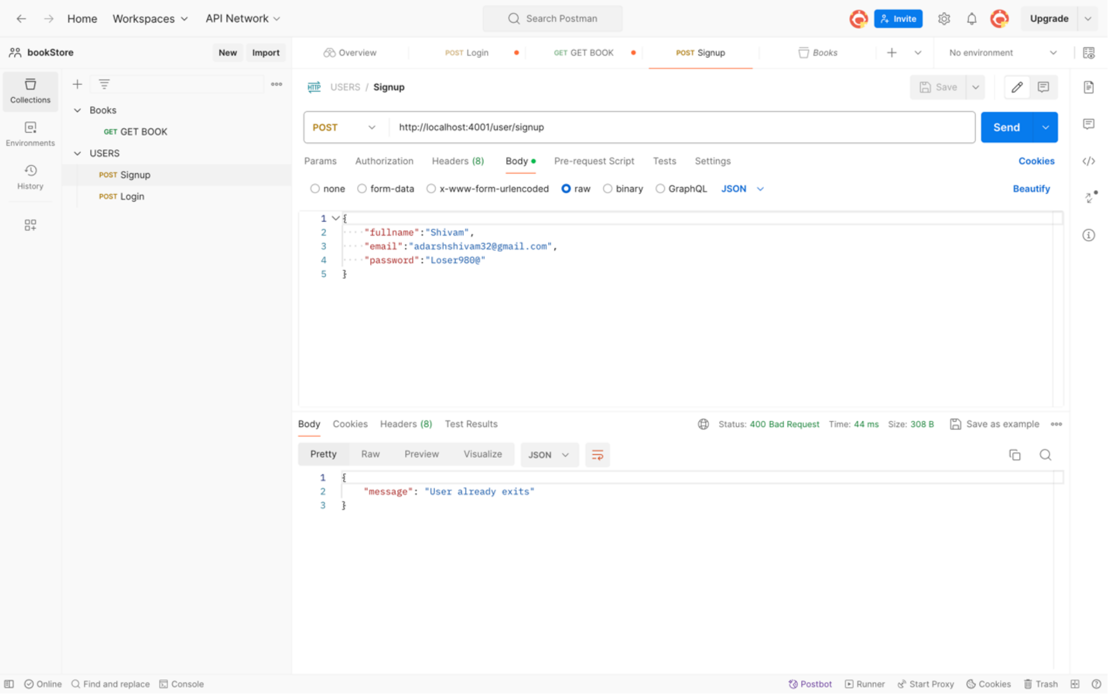
Books / GET BOOK

GET http://localhost:4001/book

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 59 ms Size: 4.43 KB

```
[
    {
        "_id": "662cea19d81e3bd8911e1370",
        "id": 1,
        "name": "Story Book",
        "title": "Lorem ipsum dolor sit amet, consectetur adipisicing elit.",
        "price": 0,
        "category": "Free",
        "image": "https://img.freepik.com/free-photo/fantasy-style-scene-international-day-education_23-2151040286.jpg?
            t=st=1714052247~exp=1714055847~hmac=b2c9f155fef4b2827838b6b03146b9c6f9645dd694dd65a426cb4def2dad3f25&w=1380"
    },
    {
        "_id": "662cea19d81e3bd8911e1371",
        "id": 2,
```



bookStore

USERS / Signup

POST http://localhost:4001/user/signup

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "fullname":"Shivam",
    "email":"adarshshivam32@gmail.com",
    "password":"Loser980@"
}
```

Body Cookies Headers (8) Test Results

Status: 400 Bad Request Time: 44 ms Size: 308 B

```
{
    "message": "User already exits"
}
```

8.Implementation

8.1. Implementation of the Project:

8.1.1. Coding: Develop the BookKidWorld website according to the design specifications and requirements outlined in earlier phases. Write clean, efficient, and maintainable code using appropriate programming languages, frameworks, and best practices.

8.1.2. Frontend Development: Implement the user interface and client-side functionality using technologies like HTML, CSS, JavaScript, React.js, and Tailwind CSS. Design and style the website to provide an intuitive and visually appealing experience for users.

8.1.3. Backend Development: Build the server-side logic, APIs, and database management using technologies like Node.js, Express.js, MongoDB, and Mongoose. Implement features such as user authentication, book/course management, and payment processing to support the functionality of the website.

8.1.4. Testing: Conduct thorough testing of the implemented features to identify and resolve any bugs, errors, or issues. Perform unit tests, integration tests, and system tests to validate the functionality, performance, and reliability of the BookKidWorld website.

8.2. Conversion Plan:

8.2.1. Migration Strategy: Develop a conversion plan to migrate existing data, users, and functionality to the new BookKidWorld platform. Define the steps and procedures for transitioning from the old system to the new system while minimizing disruption to users and stakeholders.

8.2.2. Data Migration: Transfer data from the legacy system to the new system using tools and scripts to ensure data integrity and consistency. Validate and reconcile migrated data to ensure accuracy and completeness.

8.2.3 User Training: Provide training and support to users and stakeholders to familiarize them with the new BookKidWorld platform.

Offer documentation, tutorials, and demonstrations to help users navigate the website and leverage its features effectively.

8.3. Post-Implementation and Software Maintenance:

8.3.1. Deployment: Deploy the BookKidWorld website to a production environment, making it accessible to users and stakeholders. Configure server settings, domain names, and security measures to ensure the website is secure and reliable.

8.3.2 Monitoring and Support: Monitor the performance, availability, and usage of the BookKidWorld website to identify and address any issues or bottlenecks. Provide ongoing technical support and assistance to users and stakeholders to resolve inquiries, troubleshoot problems, and optimize their experience.

8.3.3. Software Maintenance: Implement a software maintenance plan to address updates, patches, and enhancements to the BookKidWorld website. Regularly review and refine the website based on user feedback, changing requirements, and technological advancements to ensure it remains relevant and competitive.

8.3.4. Documentation and Knowledge Transfer: Document the implementation process, configurations, and setup instructions for the BookKidWorld website. Share knowledge and insights with team members, stakeholders, and future maintainers to facilitate continuity and sustainability.

By following these steps for implementation, conversion planning, and post-implementation maintenance, you can ensure the successful deployment and ongoing operation of the BookKidWorld website, providing a valuable resource for children, parents, and educators to discover and engage with children's literature and educational content.

9. Project Legacy

9.1. Current Status of the Project:

- Provide an overview of the current status of the BookKidWorld project, including its development progress, and operational performance.
- Highlight key milestones achieved, such as completion of frontend and backend development, user acceptance testing.
- Discuss any ongoing activities or initiatives related to the project, such as post-implementation support, software updates, and user feedback collection.

9.2. Remaining Areas of Concern:

- Identify any remaining challenges, issues, or areas of concern that need to be addressed to ensure the success and sustainability of the BookKidWorld project.
- This may include unresolved bugs or errors, performance bottlenecks, usability issues, security vulnerabilities, or gaps in functionality.
- Prioritize these concerns based on their impact on users, stakeholders, and the overall success of the project, and develop plans to address them in a timely manner.

9.3. Technical and Managerial Lessons Learned:

- Reflect on the technical and managerial lessons learned throughout the lifecycle of the BookKidWorld project, including successes, failures, and opportunities for improvement.
- Technical Lessons Learned:

Identify technical challenges encountered during development, such as integration complexities, performance optimization, or scalability issues.
Discuss solutions implemented to overcome these challenges, including new technologies adopted, design patterns applied, and best practices followed.

- Managerial Lessons Learned:
  Reflect on project management practices, such as planning, communication, risk management, and stakeholder engagement.
  Discuss lessons learned in areas such as resource allocation, team collaboration, decision-making, and adapting to changes in requirements or priorities.
- Continuous Improvement:
  Use the insights gained from these lessons learned to inform future projects and initiatives. Document best practices, pitfalls to avoid, and strategies for success to guide future teams and projects.
  Encourage a culture of continuous improvement within the organization by fostering open communication, knowledge sharing, and learning from both successes and failures.

10. Source Code or System Snapshots

index.js : Main page to render on the webpage

```
import express from 'express';
import dotenv from 'dotenv';
import mongoose from 'mongoose';
import cors from "cors";

import bookRoute from "./route/book.route.js";
import userRoute from "./route/user.route.js";

const app = express();
app.use(cors());
app.use(express.json());

dotenv.config();
const PORT = process.env.PORT || 4000;
const URI = process.env.MongoDBURI;

//Connect to mongodb
try{
    mongoose.connect(URI, {
        useNewUrlParser: true,
        useUnifiedTopology: true
    });
    console.log("Connected to mongoDB");
}
catch(error){
    console.log("Error: ", error);
}

//defining routes
app.use("/book", bookRoute);
app.use("/user", userRoute);

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});
```

10.1 For connecting backend with I have use Model and Controller using Mongoose and for frontend using axios I have use route module:

10.1.1. Model:

book.model.js

```
import mongoose from "mongoose";

const bookSchema = mongoose.Schema({
   name:String,
   price:Number,
   category:String,
   image:String,
   title:String
});
const Book = mongoose.model("Book", bookSchema);

export default Book;
```

10.1.2 Controller:

book.controller.js

```
import Book from "../model/book.model.js";

export const getBook = async (req, res) =>{
   try{
      const book = await Book.find();
      res.status(200).json(book);
   }
   catch(error){
      console.log("Error: ", error);
```

```
        res.status(500).json(error);
    }
};
```

10.1.3 Route

book.route.js

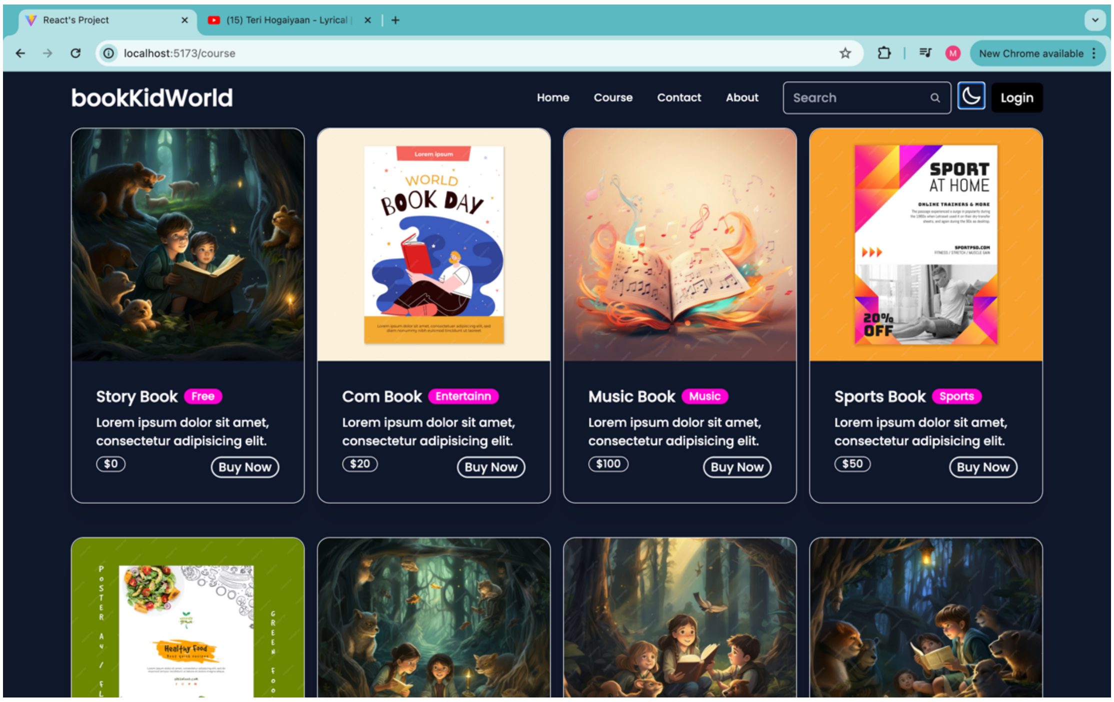
```
import express from "express";
import { getBook } from "../controller/book.controller.js";

const router = express.Router();

router.get("/", getBook);

export default router;
```

10.2 Screenshots:

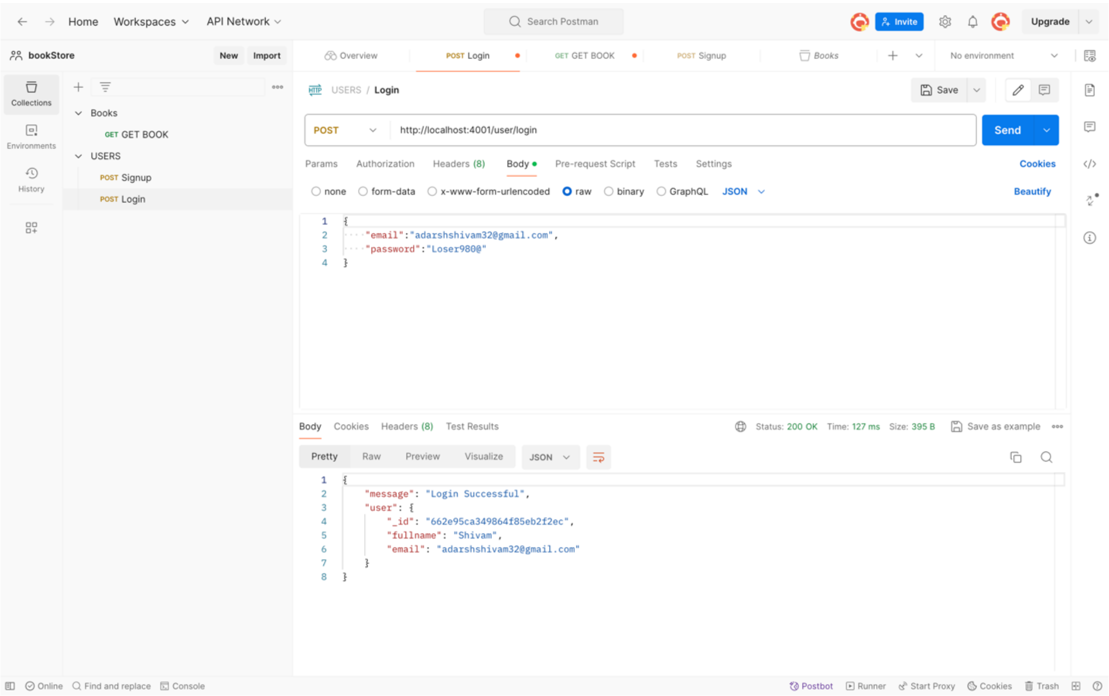


Story Book Free
Lorem ipsum dolor sit amet, consectetur adipisicing elit.
$0 Buy Now

Com Book Entertainn
Lorem ipsum dolor sit amet, consectetur adipisicing elit.
$20 Buy Now

Music Book Music
Lorem ipsum dolor sit amet, consectetur adipisicing elit.
$100 Buy Now

Sports Book Sports
Lorem ipsum dolor sit amet, consectetur adipisicing elit.
$50 Buy Now



POST http://localhost:4001/user/login

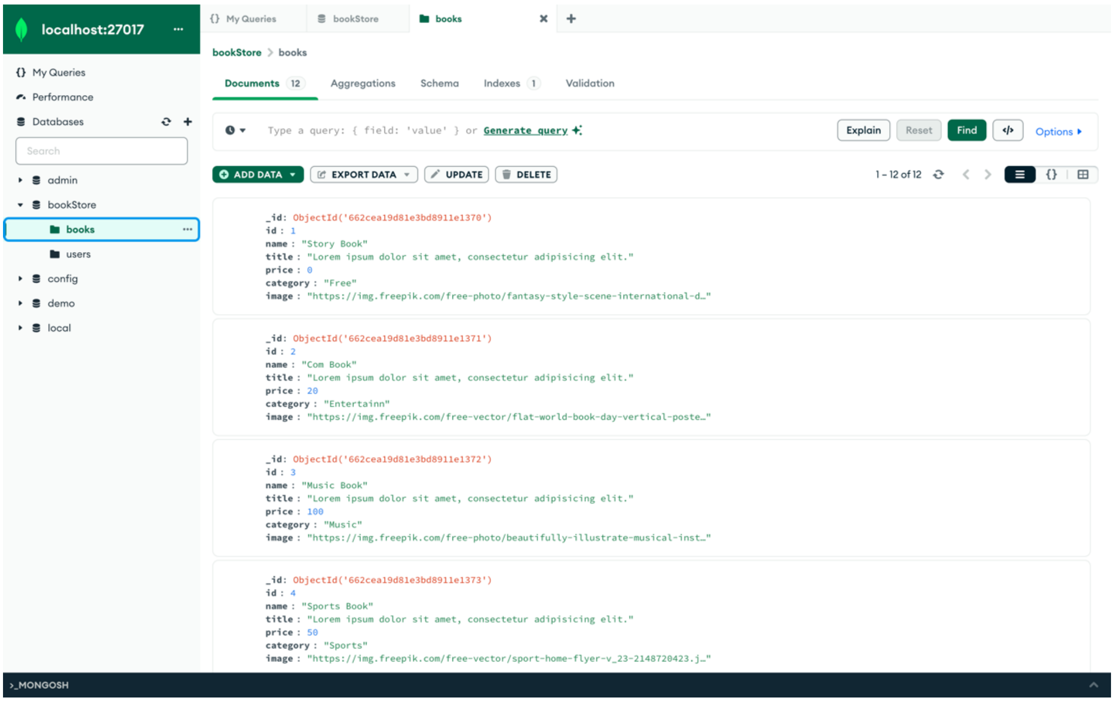
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "email":"adarshshivam32@gmail.com",
    "password":"Loser980@"
}
```

Status: 200 OK Time: 127 ms Size: 395 B

JSON

```
{
    "message": "Login Successful",
    "user": {
        "_id": "662e95ca349864f85eb2f2ec",
        "fullname": "Shivam",
        "email": "adarshshivam32@gmail.com"
    }
}
```



Type a query: { field: 'value' } or Generate query

```
_id: ObjectId('662cea19d81e3bd8911e1370')
id : 1
name : "Story Book"
title : "Lorem ipsum dolor sit amet, consectetur adipisicing elit."
price : 0
category : "Free"
image : "https://img.freepik.com/free-photo/fantasy-style-scene-international-d…"

_id: ObjectId('662cea19d81e3bd8911e1371')
id : 2
name : "Com Book"
title : "Lorem ipsum dolor sit amet, consectetur adipisicing elit."
price : 20
category : "Entertainn"
image : "https://img.freepik.com/free-vector/flat-world-book-day-vertical-poste…"

_id: ObjectId('662cea19d81e3bd8911e1372')
id : 3
name : "Music Book"
title : "Lorem ipsum dolor sit amet, consectetur adipisicing elit."
price : 100
category : "Music"
image : "https://img.freepik.com/free-photo/beautifully-illustrate-musical-inst…"

_id: ObjectId('662cea19d81e3bd8911e1373')
id : 4
name : "Sports Book"
title : "Lorem ipsum dolor sit amet, consectetur adipisicing elit."
price : 50
category : "Sports"
image : "https://img.freepik.com/free-vector/sport-home-flyer-v_23-2148720423.j…"
```

11. Bibliography

1. Postman Documentation. Available at:
https://learning.postman.com/docs/

2. React.js Documentation. Available at: https://reactjs.org/docs/getting-started.html
3. MongoDB Documentation. Available at: https://docs.mongodb.com/
4. Tailwind CSS Documentation. Available at: https://tailwindcss.com/docs
5. Node.js Documentation. Available at: https://nodejs.org/en/docs/
6. "Best Educational Websites for Kids." Common Sense Media. https://www.commonsense.org/education/top-picks/best-educational-websites-for-kids
7. "Node.js Express API Development Tutorial." MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs
8. "React.js Tutorial for Beginners." FreeCodeCamp. https://www.freecodecamp.org/news/react-tutorial-for-beginners/
9. "MongoDB Atlas Documentation." MongoDB. https://docs.mongodb.com/manual/
10. "Tailwind CSS Cheat Sheet." Tailwind Toolbox. https://www.tailwindtoolbox.com/cheatsheet
11. "Postman API Testing Guide." Postman Learning Center. https://learning.postman.com/docs/publishing-your-api/testing-your-api/
12. "Web Development Resources for Kids." Mozilla Developer Network (MDN). https://developer.mozilla.org/en-US/docs/Learn
13. "Best Practices for Node.js Development." Node.js Foundation. https://nodejs.org/en/docs/guides/
14. https://daisyui.com/ Component library for tailwind CSS.