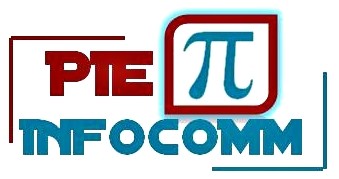
****

**SESSION – 2024-2025**

**Final Project Synopsis**

**On**

**Create X (DIY Platform) For Sharing and Discovering Do It Yourself Ideas**

*Submitted in fulfillment of requirement of the internship of*

**Full Stack Web Development using MERN Stack**

*To*

**PIE INFOCOMM PVT. LTD**.

*by*

**Kushagra Chaturvedi**

*Under the guidance of*

**Ms. Aishwarya Saxena**

**(Sr. Software Developer)**

**TABLE OF CONTENTS**

1. Introduction
2. Objective
3. Background
4. Hardware and Software Requirement
5. ERD and DFD
6. Form Layout
7. Database Schema
8. Future Scope
9. Conclusion

References and Bibliography

**INTRODUCTION**

The DIY (Do It Yourself) project platform is a web-based application designed to offer a comprehensive space for enthusiasts to share and discover DIY project ideas and tutorials. Developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), it harnesses modern web technologies to provide a seamless and engaging user experience. Users can create accounts, post their DIY projects, browse through an extensive collection of others’ projects, comment, and interact with fellow DIY enthusiasts. This platform promotes creativity and knowledge sharing, fostering a community of like-minded individuals passionate about DIY endeavors.

At its core, the platform aims to inspire and empower users by providing access to a diverse range of projects across various categories, from simple home decor to advanced electronics. Each project includes detailed instructions, materials needed, and step-by-step guides, ensuring users of all skill levels can successfully complete their chosen projects. The interactive features, such as commenting and project sharing, encourage collaboration and feedback, enriching the learning experience. By bringing together a vibrant community of DIY enthusiasts, the platform not only facilitates the exchange of ideas and techniques but also helps build lasting connections among creators.

**OBJECTIVE**

The primary objective of this project is to develop a robust and user-friendly web application that empowers users to share and discover DIY projects effortlessly. This platform aims to foster a community of creativity and knowledge sharing by providing a seamless and engaging experience for users. The specific goals include:

- **Facilitate Secure User Registration and Authentication:** Implement a secure system for user registration and authentication, ensuring that users can safely create accounts and access their profiles. This includes handling user data with encryption and following best practices for data security.

- **Enable CRUD Operations for DIY Projects:** Allow users to create, read, update, and delete DIY project posts. This functionality ensures that users can manage their content easily and keep their project information up-to-date.

**- Browse and Search DIY Projects by Categories and Keywords:** Provide robust browsing and searching capabilities, enabling users to find DIY projects based on categories and keywords. This feature will help users quickly locate projects that match their interests and needs.

**- Commenting System for User Interaction and Feedback:** Develop a commenting system that allows users to interact with each other and provide feedback on projects. This promotes community engagement and helps users learn from each other’s experiences.

**- Ensure Responsive Design for Seamless Experience Across Devices:** Design the platform with a responsive layout that works seamlessly across various devices, including desktops, tablets, and smartphones. This ensures that users have a consistent and user-friendly experience regardless of the device they use.

**- Utilize the MERN Stack for Modern Web Development:** Leverage the strengths of the MERN stack (MongoDB, Express.js, React.js, Node.js) to build a modern and efficient web application. Each component of the MERN stack offers unique benefits, such as the flexibility of MongoDB, the robust routing capabilities of Express.js, the dynamic and responsive user interface of React.js, and the efficient server-side handling of Node.js.

By achieving these objectives, the platform will serve as a comprehensive resource for DIY enthusiasts, providing a space where they can share their creativity, discover new projects, and connect with a community of like-minded individuals.

**BACKGROUND**

The concept of DIY projects has seen a remarkable rise in popularity in recent years, driven by an increasing interest in self-sufficiency, creativity, and cost-effectiveness. As more individuals look to personalize their spaces, develop new skills, and save money, the DIY movement has grown into a vibrant community. DIY platforms serve as a critical resource in this trend, offering users a vast array of projects that span various categories such as home improvement, gardening, arts and crafts, electronics, and more. These platforms enable users to not only find inspiration and ideas but also access detailed tutorials and guides to help them complete their projects successfully.

Traditional DIY platforms, however, often face limitations in terms of user engagement and modern features. Many existing platforms struggle with outdated interfaces, lack of interactive elements, and limited functionality for user interaction and content management. This is where leveraging the MERN stack (MongoDB, Express.js, React.js, Node.js) offers a significant advantage. The MERN stack provides a comprehensive suite of technologies that collectively enhance the functionality, responsiveness, and overall user experience of the platform.

* **MongoDB**

MongoDB's flexible, schema-less data structure is ideal for handling the diverse and complex data associated with DIY projects. Its ability to store data in JSON-like documents makes it easy to manage various project details, user profiles, comments, and other related information. MongoDB’s scalability and performance are crucial for ensuring that the platform can handle a growing number of users and an expanding database of DIY projects.

* **Express.js and Node.js**

On the backend, Express.js and Node.js offer robust and efficient handling of server-side operations. Express.js simplifies the process of creating and managing routes, handling HTTP requests, and integrating with databases, while Node.js provides a powerful runtime environment for executing server-side code. Together, they form a solid foundation for building a fast, secure, and scalable backend. This combination enables seamless user authentication, real-time data updates, and efficient handling of CRUD (Create, Read, Update, Delete) operations for DIY projects.

* **React.js**

React.js is essential for developing a dynamic and responsive frontend. Its component-based architecture allows for the creation of reusable UI components, which not only enhances development efficiency but also ensures a consistent user interface across the platform. React’s virtual DOM significantly improves the performance of dynamic web applications, allowing users to interact with the platform smoothly and efficiently. Features like real-time project updates, interactive commenting, and responsive design are easily implemented with React.js, ensuring a rich and engaging user experience.

By integrating these modern web development technologies, the DIY platform can overcome the limitations of traditional platforms and offer a more interactive, user-friendly, and scalable solution. This approach not only meets the current needs of DIY enthusiasts but also positions the platform to adapt and grow with future trends and user demands.

**HARDWARE AND SOFTWARE REQUIREMENTS**

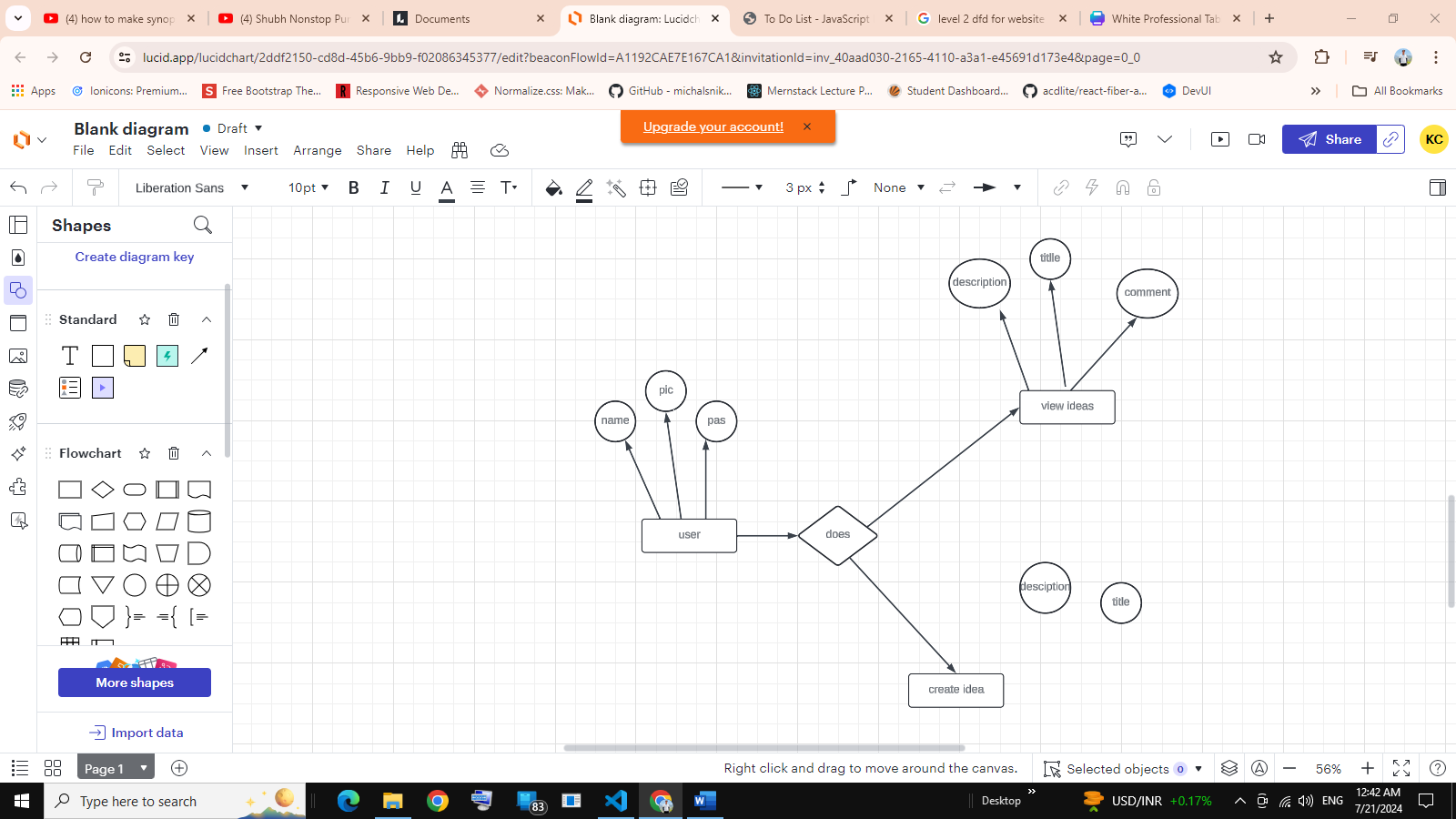
* **Hardware Requirements:**

|  |  |
| --- | --- |
| Minimum Ram | 8 GB DDR4 |
| Processor | i5 or equivalent |
| System Type | 64-bit Operating System |
| Windows Version | Window 8 or above |

* **Software Requirements:**

|  |  |
| --- | --- |
| Frontend Tools | HTML5 , CSS3, JavaScript |
| Frontend Scripting Language | ReactJS |
| Backend Scripting Language | NodeJs , ExpressJs |
| Database | MongoDB Atlas |
| Code Editor | VS CODE |

**ERD AND DFD**

Entity-Relationship Diagram (ERD):

**Data Flow Diagram (DFD):**

1. LEVEL – 0

A screenshot of a computer

Description automatically generated

1. LEVEL – 1

A screenshot of a computer screen

Description automatically generated

1. LEVEL–2

Check Roles of Access







Manage Blog Details

Manage Blog Details

Login

Admin

Manage Modules

Check Credentials

Create Account

Manage Blogs

Manage Blogs

Id , Password Created

**FORM LAYOUTS**

The form layouts define the user interface for various functionalities within the application. Key forms include:

**- Registration Form:** Captures user information such as username, email, and password.

LOGO

SIGNUP

User Name

Password

SIGNUP

Already have an account ? Login

**- Login Form:** Authenticates users with their email and password.

LOGIN

User Name 

Password

LOGIN

New here? Register

LOGO

New Here ? Register

Login

**- Idea Submission Form:** Allows users to submit new DIY projects, including fields for title, description, and image upload.

PROFILE

ALL BLOGS MY BLOGS CREATE

LOGO

**Create Your Idea**

Title

Description

Choose File

Upload Photo

Create Now

**- Comment Form:** Enables users to add comments to projects.

PROFILE

LOGO

ALL BLOGS MY BLOGS CREATE

TITLE

DESCRIPTION

IDEA PHOTO

XYZ

XYZ

COMMENTS

ADD COMMENT

Your comment……..

Add a comment

**DATABASE SCHEMA**

The database schema outlines the structure of the data stored in MongoDB. Key collections include:

**- Users:** Stores user information such as username, email, and hashed password.

**- Ideas** Contains project details including title, description, image URL, and references to the user who created it.

**- Comments:** Holds comments related to projects, including the content of the comment and references to the user and project.

**FUTURE SCOPE**

The future scope of the DIY project platform is vast and promising, with numerous enhancements and expansions planned to further enrich the user experience and ensure the platform remains a vibrant and sustainable community for DIY enthusiasts. Key areas of development include:

1. **User Profiles**

Expanding user profiles is a critical next step for enhancing user engagement and personalization. By allowing users to create comprehensive profiles, the platform can offer features such as:

- **Portfolio of Projects:** Users can showcase a portfolio of their completed and ongoing projects, which not only serves as a personal achievement gallery but also inspires other users. This feature can include detailed project descriptions, images, and even video tutorials.

- **Follower/Following Features:** Implementing follower/following capabilities will enable users to connect with each other more meaningfully. Users can follow their favorite creators, receive updates on new projects, and build a network of like-minded DIY enthusiasts. This social aspect will foster a sense of community and encourage regular platform engagement.

1. **Mobile Application**

Developing a mobile application for iOS and Android is essential to cater to the growing number of users who prefer accessing content on their mobile devices. A mobile app will provide the following benefits:

- **On-the-Go Access:** Users can browse, create, and manage their DIY projects from anywhere, making the platform more convenient and accessible.

- **Enhanced User Experience:** A native mobile app can offer a more tailored and responsive user experience, utilizing device-specific features such as push notifications, camera integration for project documentation, and offline access to project guides.

- **Broader Reach:** A mobile app will help the platform reach a wider audience, including those who primarily use smartphones and tablets for their online activities.

1. **Integration with Social Media**

Integrating the platform with popular social media networks will significantly increase its visibility and user engagement. Key integration features include:

**- Easy Sharing:** Users can share their projects directly to social media platforms like Facebook, Instagram, Pinterest, and Twitter. This will not only promote individual projects but also drive traffic back to the platform.

**- Social Media Logins:** Allowing users to sign in with their social media accounts will streamline the registration process and encourage more sign-ups.

**- Engagement Metrics:** Social media integration can provide valuable insights into how projects are being received and shared across different platforms, helping to refine content strategies and user engagement tactics.

1. **Monetization**

Introducing monetization strategies is crucial for sustaining the platform in the long term. Potential revenue models include:

**- Premium Features:** Offering premium features such as ad-free browsing, exclusive project tutorials, and advanced project management tools can generate subscription-based revenue. Premium users could also access features like direct messaging with project creators and personalized project recommendations.

**- Ad-Based Revenue:** Implementing targeted advertising can provide a steady revenue stream without compromising the user experience. Ads related to DIY supplies, tools, and relevant services can be integrated seamlessly into the platform.

**- Affiliate Marketing:** Partnering with DIY supply vendors and including affiliate links in project guides can generate additional income. When users purchase supplies through these links, the platform earns a commission.

**- Sponsored Content**: Collaborating with brands for sponsored projects and tutorials can offer another revenue stream. Brands can gain exposure to a targeted audience, while users benefit from high-quality content and potential discounts on supplies.

By pursuing these future developments, the DIY project platform can evolve into a comprehensive and thriving ecosystem for DIY enthusiasts. The enhancements in user profiles, the addition of a mobile application, social media integration, and strategic monetization efforts will not only improve the user experience but also ensure the platform's long-term sustainability and growth.

**CONCLUSION**

The DIY project platform represents a significant advancement in fostering an engaging and interactive community for DIY enthusiasts. By leveraging the power of the MERN stack, the platform is designed to be scalable, efficient, and highly responsive, offering a seamless user experience that can be continuously enhanced to meet evolving user needs. This technological foundation ensures that the platform is not only robust in handling the diverse and dynamic data associated with DIY projects but also flexible enough to adapt to future innovations and user demands.

One of the platform’s core strengths is its ability to create a vibrant space where users can share their creativity and passion for DIY projects. The platform offers a rich repository of project ideas and tutorials across a wide range of categories, catering to both novice and experienced DIYers. By enabling users to create, read, update, and delete project posts, the platform ensures that content remains fresh and relevant. Additionally, the browsing and searching capabilities allow users to quickly find projects that match their interests, making the platform a valuable resource for inspiration and practical guidance.

The integration of a commenting system further enhances the user experience by promoting interaction and feedback. Users can engage in discussions, share tips, and provide support to one another, fostering a collaborative and supportive community. This interactive element is crucial for building a sense of belonging and encouraging users to actively participate and contribute to the platform.

Looking ahead, the planned future enhancements will significantly expand the platform’s capabilities and user base. The development of comprehensive user profiles, a mobile application, and social media integration will not only improve user engagement and accessibility but also extend the platform’s reach and influence. These features will enable users to connect more deeply with the community, showcase their work, and access the platform from any device, at any time.

Moreover, the introduction of monetization strategies such as premium features, ad-based revenue, affiliate marketing, and sponsored content will ensure the platform’s long-term sustainability. These revenue models will provide the necessary financial support to maintain and grow the platform, allowing for continuous improvements and the addition of new features.

**REFERENCES AND BIBLIOGRAPHY**

- MongoDB Documentation. - Retrieved from <https://docs.mongodb.com>

- Express.js Documentation. - Retrieved from <https://expressjs.com>

- React.js Documentation. - Retrieved from <https://reactjs.org>

- Node.js Documentation. - Retrieved from <https://nodejs.org>

- JWT Documentation - Retrieved from <https://jwt.io>