***Inn Vantage: Reduce paper work with tech work.***

**Synopsis**

**On**

**CODING WEBSITE**

# to be developed to fulfil the requirements for

**3rd Year Project (CSE-2024)**

Submitted to

Department of Computer Science & Engineering

Chitkara University, Punjab



Under the supervision of -

Mrs. Suhaib Khan

Submitted by

Shubham Jain 2210992365

Shubham Verma 2210992371

Shubham Bansal 2201992361

Shubham Thakur 2210992371

Shubham 2210992360

**Table of Content**

|  |  |
| --- | --- |
| **Sr. No.** | **Name** |
| **1.** | Abstract |
| **2.** | **Introduction to the project** |
| **3.** | **Software and Hardware Requirement Specification** |
| **4.** | Hypothesis |
| **5.** | Methodology |
| **6.** | Technology & Tools |
| **7.** | Conclusion |
| **8.** | Future Ideas |

## (Annexure -E)

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of study.

We express our sincere gratitude to all for providing me an opportunity to undergo Integrated Project as the part of the curriculum.

Lastly, We would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to day experience and received lots of suggestions that improve our quality of work.

**1.Abstract**

The project titled "Kido Code" is a comprehensive web-based platform designed to streamline and enhance the educational experience by integrating various stakeholders within an academic environment, including management authorities, teachers, and students.

Kido Code is a web-based platform specifically designed for children aged 8 to 14, providing a beginner-friendly and interactive introduction to coding. The platform is built on the philosophy of "learning by doing," where children are encouraged to actively engage with coding concepts through hands-on activities. Kido Code combines educational content with interactive tools such as a code editor and multiple-choice quizzes (MCQs) to create an engaging and fun learning environment.

The platform is divided into various modules and courses, each tailored to different age groups and skill levels. The aim is to introduce coding in a gradual manner, starting with basic concepts like variables, loops, and conditionals, and progressing towards more advanced topics such as functions, data structures, and algorithms.

By blending education with entertainment, Kido Code encourages kids to explore coding concepts in a way that fosters creativity, logical thinking, and problem-solving skills. The platform also incorporates elements of gamification—such as badges, leaderboards, and rewards—to motivate children to continue learning and improving their coding abilities.

In conclusion, Kido Code is a significant step forward in the digital transformation of educational institutions, providing a scalable solution that can be customized to meet the specific needs of different academic environments. The platform not only simplifies administrative tasks but also enhances the overall educational experience for both teachers and students, ultimately contributing to better academic outcomes.

**2. Introduction to the project**

Kido Code is an online educational platform designed to teach children the basics of coding in a fun, interactive, and engaging way. In today's digital age, coding has become an essential skill for the future workforce. Kido Code aims to bridge the gap by providing young learners with the tools and resources they need to develop a strong foundation in coding. This report provides a comprehensive overview of the platform, detailing its features, development process, and future directions.

**2.1 Background**

The landscape of education has undergone significant changes over the past few decades, driven largely by advancements in technology and the increasing demand for more efficient, scalable, and inclusive learning environments. Educational institutions, ranging from schools to universities, are under pressure to adopt digital solutions that can streamline administrative tasks, improve communication, and enhance the overall learning experience. However, the adoption of such technologies has often been fragmented, with different systems addressing specific needs without a cohesive framework that integrates all aspects of academic management.

The primary objectives of Kido Code are as follows:

* **Introduce Fundamental Coding Concepts**: Teach basic programming constructs such as variables, loops, conditionals, and functions through interactive exercises and projects.
* **Create an Engaging Learning Experience**: Use gamification and interactive quizzes to keep learners motivated and engaged.
* **Promote Hands-on Learning**: Encourage experimentation and practice through a built-in code editor, allowing kids to write, run, and debug code in real-time.
* **Provide a Safe Learning Environment**: Ensure that the platform is safe, secure, and age-appropriate, with strict privacy controls and content moderation.
* **Encourage Continuous Learning**: Track progress and reward achievements to motivate learners to continue developing their coding skills.

**2.2 Problem Statement**

In today’s rapidly evolving digital world, coding has become an essential skill for future generations. However, there is a significant gap in accessible, engaging, and age-appropriate coding education for children, particularly those aged 8 to 14. Traditional educational systems often lack the resources, expertise, and curriculum to effectively teach coding to young learners. As a result, many children miss out on the opportunity to develop critical digital literacy skills at an early age, which are increasingly important for future academic and career success.

1. **Software and Hardware Requirement Specification**

The "Kido Code" platform is built using a modern stack that ensures scalability, efficiency, and ease of development. The core technologies include React.js for the frontend, Node.js with Express.js for the backend, and MongoDB for database management. Below is a detailed breakdown of the software specifications:

**1. Frontend:**

* **React.js:**
  + A powerful JavaScript library used for building user interfaces, particularly single-page applications.
  + It enables the development of reusable UI components, which enhances the development process and improves maintainability.
  + React Router is used for managing in-app navigation.

**2. Backend:**

* **Node.js:**
  + A JavaScript runtime built on Chrome's V8 engine, used for executing server-side code.
  + It allows the use of JavaScript for both frontend and backend, providing a unified development experience.
* **Express.js:**
  + A fast, unopinionated, and minimalist web framework for Node.js.
  + Used for building RESTful APIs, handling HTTP requests, and managing middleware.
  + Supports various middleware to handle authentication, logging, error handling, etc.

**3. Database:**

* **MongoDB:**
  + A NoSQL database that stores data in JSON-like documents with flexible schemas.
  + Ideal for applications requiring a flexible, scalable database solution.
  + Mongoose, an Object Data Modeling (ODM) library, is used for MongoDB to manage data relationships and enforce schemas.

**4. Other Tools and Technologies:**

* **Axios:**
  + A promise-based HTTP client for making HTTP requests from the browser and Node.js.
  + It is used for communicating between the frontend (React.js) and backend (Express.js).
* **JSON Web Tokens (JWT):**
  + For secure authentication and session management across the application.
* **Nodemon:**
  + A tool that helps develop Node.js applications by automatically restarting the server when file changes are detected.

3**.1 Programming/Working Requirements**

The "Kido Code" project is developed within a well-structured programming and working environment that enhances productivity, collaboration, and code quality. The primary tool used for development is **Visual Studio Code (VS Code)**, a highly popular and versatile code editor that supports a wide range of programming languages and extensions.

1. **Development Environment:** Visual Studio Code (VS Code):
   * + **Code Editor:** VS Code is the primary code editor used for writing and managing the source code of the "Academia" project.

Node.js & npm:

* + - Runtime Environment**:** Node.js serves as the runtime environment for executing JavaScript code on the server-side.
    - Package Manager**:** npm (Node Package Manager) is used for managing project dependencies, scripts, and packages required for the project

**2. Development Workflow:**

* **Integrated Development:**
  + VS Code allows seamless integration between the frontend (React.js) and backend (Node.js with Express.js), enabling developers to work on different aspects of the project within a single environment.
* **Debugging:** VS Code provides robust debugging tools for both client-side and server-side code, allowing developers to set breakpoints, inspect variables.

**3.2 Requirements to Run the Application:**

To successfully run the "Kido Code" application, several software and hardware requirements must be met. These requirements ensure that the application performs optimally in a development, testing, or production environment. Below are the key requirements:

**1. Software Requirements:**

**Frontend:**

* **Web Browser:**
  + A modern web browser such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari. The application is optimized for the latest versions of these browsers.
* **Node.js and npm:**
  + **Node.js (version 14.x or higher):** Required to run the React development server and build the frontend application.
  + **npm (Node Package Manager):** Used to manage dependencies and run scripts for building and serving the React application.

**Backend:**

* **Node.js (version 14.x or higher):**
  + Required to run the Express.js server that powers the backend of the application.
* **Express.js:**
  + A Node.js framework used to build the RESTful API for the backend.
* **MongoDB:**
  + **MongoDB Server (version 4.x or higher):** Required to store and manage application data. This can be a locally installed instance or a cloud-based service like MongoDB Atlas

1. **Database Analysing:**

The "Kido Code" project utilizes MongoDB as its database management system. MongoDB is a NoSQL database that stores data in JSON-like documents, providing flexibility in schema design and scalability. Below is a detailed overview of how the database is analyzed, designed, and implemented for the project:

**1. Database Analysis:**

* **Requirements Gathering:**
  + The first step involves understanding the key entities and relationships within the "Academia" system. These include users (management, teachers, students), courses, assignments, and academic records.
  + Analysis of the data flow and interactions between these entities is crucial for designing an efficient and normalized database schema.
* **Use Cases Identification:**
  + Identify the primary use cases, such as user registration, course enrollment, assignment submissions, and performance tracking. These use cases inform the structure and relationships within the database.

1. **System Testing**

System testing for the "Academia" platform involves verifying that all components of the application work together as expected. This includes testing the integration of frontend and backend, database operations, and the API endpoints. Postman plays a crucial role in testing the backend APIs to ensure they perform correctly under various conditions.

**1. API Testing with Postman:**

* **Endpoint Validation:**
  + Use Postman to send HTTP requests (GET, POST, PUT, DELETE) to your API endpoints to ensure they return the expected responses.
  + Test the CRUD operations (Create, Read, Update, Delete) for various entities like users, courses, assignments, and submissions.
  + Example: Sending a GET request to /api/courses to retrieve a list of courses and verify the response structure and data.
* **Data Validation:**
  + Validate that the data returned by the API matches the expected output, including correct status codes, data formats, and error messages.
  + Example: Testing the response of a POST request to /api/users to ensure that user registration is handled correctly and that any validation errors are appropriately returned.
* **Authentication and Authorization Testing:**
  + Test secure endpoints that require authentication using JWT tokens. Ensure that unauthorized requests are rejected and that authorized requests return the correct data.
  + Example: Using Postman to include a JWT token in the Authorization header for a request to /api/assignments to verify that only authenticated users can access the data.
* **Performance Testing:**
  + Perform stress testing by sending multiple requests to an endpoint to ensure the API can handle high loads and respond within acceptable time limits.
  + Example: Using Postman’s Runner feature to simulate multiple concurrent users accessing the /api/courses endpoint.
* **Environment Testing:**
  + Create different environments in Postman (e.g., development, testing, production) and test API responses across these environments to ensure consistency and correct configuration.

2.**Integration Testing:**

* Ensure that the frontend can successfully communicate with the backend API. Verify that data flows correctly between the client-side React application and the server-side Node.js/Express API.
* Example: Testing the user registration process from the frontend form submission to the backend database insertion and ensuring the correct feedback is provided to the user.

**3. End-to-End Testing:**

* Conduct end-to-end tests to simulate real-world scenarios, such as a student enrolling in a course, submitting an assignment, and receiving a grade.
* Example: Using Postman to automate the sequence of API calls that represent a complete user journey, verifying the correct interactions between components.

1. **Database Testing:**

* Test MongoDB operations by verifying that data is correctly inserted, updated, retrieved, and deleted as expected through the application.
* Example: After performing a POST request to add a new course, check MongoDB to ensure the course data is stored correctly.

1. **Conclusion**

The Kido Code platform is a comprehensive solution designed to bridge the gap in coding education for children aged 8 to 14. In an era where digital literacy is becoming increasingly vital, Kido Code provides a unique, engaging, and effective approach to teaching coding. The platform is built around the core principles of interactivity, safety, personalization, and hands-on learning, ensuring that young learners not only acquire essential coding skills but also develop a love for learning and problem-solving.

**Key Achievements**

1. **Engaging and Interactive Learning Experience**: Kido Code successfully addresses the challenge of engaging young learners by integrating a variety of interactive tools, including a visual code editor and gamified quizzes. By transforming the learning process into an enjoyable experience, the platform ensures that children remain motivated and interested in exploring coding concepts. The incorporation of gamification elements such as badges, leaderboards, and rewards further enhances engagement, creating a sense of accomplishment and encouraging continuous learning.
2. **Foundational Skill Development**: The platform focuses on building a strong foundation in coding by starting with basic programming constructs and gradually introducing more complex concepts. This progressive approach ensures that children develop a deep understanding of fundamental coding principles, which is crucial for their future success in more advanced programming courses. By emphasizing critical thinking, logical reasoning, and problem-solving, Kido Code also fosters essential cognitive skills that go beyond coding and benefit overall academic growth.
3. **Safe and Supportive Environment**: Kido Code prioritizes the safety and security of its young users. With robust privacy controls, parental monitoring features, and moderated content, the platform provides a secure and supportive environment for children to learn. This focus on safety helps build trust among parents and caregivers, who are reassured that their children are engaging in a positive and protected online learning experience.
4. **Personalized Learning Paths**: Recognizing that every child learns differently, Kido Code offers personalized learning paths that adapt to individual needs and progress. By tracking user performance and providing tailored feedback and recommendations, the platform ensures that each learner can progress at their own pace. This adaptive learning model helps maintain a child's interest and motivation while providing a customized educational experience that caters to their unique strengths and areas for improvement.
5. **Real-World Applications and Practice Opportunities**: Kido Code’s emphasis on practical, hands-on learning through its built-in code editor and real-time coding challenges ensures that children can immediately apply what they learn. By engaging in real-world scenarios and projects, learners understand the practical relevance of coding skills, which enhances their appreciation and retention of the material. This approach not only reinforces theoretical knowledge but also builds confidence and competence in applying coding skills to solve real-life problems.
6. **Future Scope**

While Kido Code has made significant strides in providing a valuable educational tool for young learners, there are several opportunities for future enhancements:

**1. Mobile Application Development:**

* Cross-Platform Mobile App:
  + Developing a mobile application for both Android and iOS platforms to provide users with on-the-go access to the Academia platform. A mobile app would enhance accessibility, especially for students and teachers who need to interact with the platform outside traditional desktop environments.

**2. Advanced Security Features:**

* Two-Factor Authentication (2FA):
  + Implementing two-factor authentication to enhance user security, ensuring that only authorized users can access the platform.
* Data Encryption:
  + Upgrading the platform’s security infrastructure to include advanced encryption techniques for data storage and transmission, protecting sensitive information from breaches.

**3. Data Analytics and Reporting:**

* Comprehensive Reporting Tools:
  + Developing advanced data analytics and reporting tools that provide insights into academic performance, resource utilization, and student engagement. These tools could generate custom reports for management, teachers, and students.
* Dashboard Customization**:**
  + Allowing users to customize their dashboards with widgets that display relevant analytics, helping them track key metrics and make informed decisions.