



AI Website Builder Using Pern Stack

Prathamesh Jalindar Yadav¹, Vishwajeet Milind Chavan², Mrs. S.S.Parekh³

Students, Department of Computer Science & Engineering,

Padmabhooshan Vasatraodada Patil Institute of Technology (PVPIT), Budhgaon (Sangli), India^{1,2}

Guide, Department of Computer Science & Engineering,

Padmabhooshan Vasatraodada Patil Institute of Technology (PVPIT), Budhgaon (Sangli), India³

Abstract: The AI Powered Website Builder is a smart web application designed to help users generate websites automatically using Artificial Intelligence. The system uses the PERN Stack (PostgreSQL, Express.js, React.js, and Node.js) for full-stack development, Neon Database for cloud-based PostgreSQL storage, and OpenRouter API for integrating multiple AI models. The platform allows users to enter prompts describing their desired website, after which the AI generates website structure, content, and design suggestions automatically. This project reduces manual website development effort, improves productivity, and enables non-technical users to create websites easily. The system provides fast generation, responsive UI, cloud database storage, and scalable architecture

Keywords: Artificial Intelligence, AI Website Builder, PERN Stack, React.js, Node.js, Express.js, PostgreSQL, Neon Database, OpenRouter API, Better Auth, Full Stack Development, Website Generation, Cloud Database, Generative AI, Web Automation, Responsive Web Design.

I. INTRODUCTION

Website development traditionally requires knowledge of frontend design, backend programming, and database management. Many small businesses, students, and beginners face difficulties creating professional websites due to lack of technical skills and high development costs.

With the advancement of Artificial Intelligence and Large Language Models (LLMs), website generation can now be automated using natural language prompts. AI-powered systems can understand user requirements and generate layouts, content, and code automatically.

The proposed AI Website Builder uses the PERN stack for application development and integrates the OpenRouter API to access AI models for content and website generation. Neon Database is used as a cloud-hosted PostgreSQL database for storing user data, generated projects, prompts, and configurations.

The system simplifies website development by enabling users to generate websites through simple text instructions, making website creation faster, smarter, and more accessible.

II. OBJECTIVES

1. To develop an AI-based website builder using the PERN stack architecture.
2. To integrate OpenRouter API for AI-powered content and layout generation.
3. To store user projects and generated website data using Neon PostgreSQL database.
4. To provide a responsive and user-friendly interface for website creation.
5. To reduce manual coding effort using prompt-based automation.
6. To allow users to generate website templates dynamically.
7. To improve website development speed and accessibility for beginners.

III. MODULES OF PROJECT

Module 1: User Authentication

- User signup and login system is developed.
- BETTERAUTH authentication is used for secure access.
- User sessions and credentials are managed securely.

Module 2: AI Prompt Processing

- Users enter website requirements in text form.



- Prompt is sent to OpenRouter API.
- AI model processes the request and generates website structure/content.

Module 3: Website Generation Module

- Generated code and layouts are displayed dynamically.
- React components render generated website pages.
- Users can preview generated websites instantly.

Module 4: Database Management

- Neon PostgreSQL database stores:
 - User details
 - Website prompts
 - Generated projects
 - History records
- Express.js APIs manage database operations.

Module 5: Frontend Development

- React.js frontend provides interactive UI.
- Responsive design improves usability.
- Users can edit generated content dynamically.

Module 6: Backend API Services

- Node.js and Express.js handle server operations.
- APIs manage AI communication and database requests.
- Backend ensures secure and scalable processing.

IV. REQUIREMENT ANALYSIS

Name of Equipment	Specification	Cost
Laptop/Desktop	Intel i5, 8GB RAM, 512GB SSD	50000
Operating System	Windows 10/11 or Linux	-
Frontend Framework	React.js	Free
Backend Framework	Node.js + Express.js	Free
Database	Neon PostgreSQL	Free/Paid
AI Integration	OpenRouter API	Based on usage
Code Editor	VS Code	Free
Browser	Chrome/Edge	Free
Deployment Platform	Vercel/HOSTINGER	Free/Paid

Table No: 4.1



V.ARCHITECTURE ND WORKFLOW

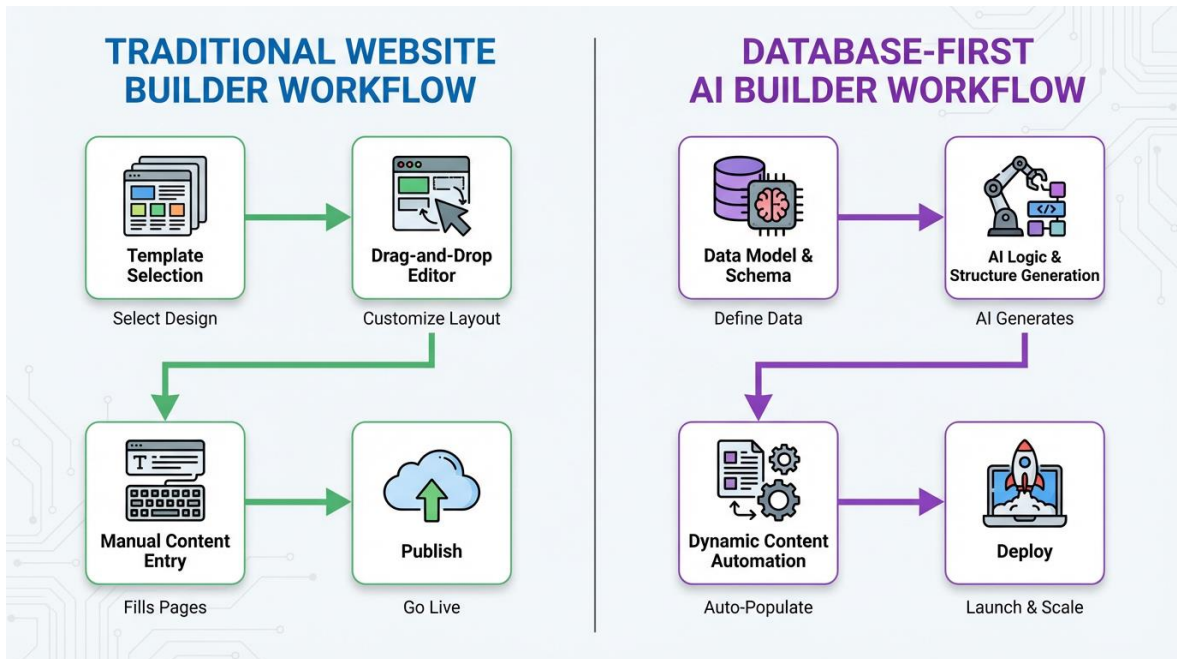


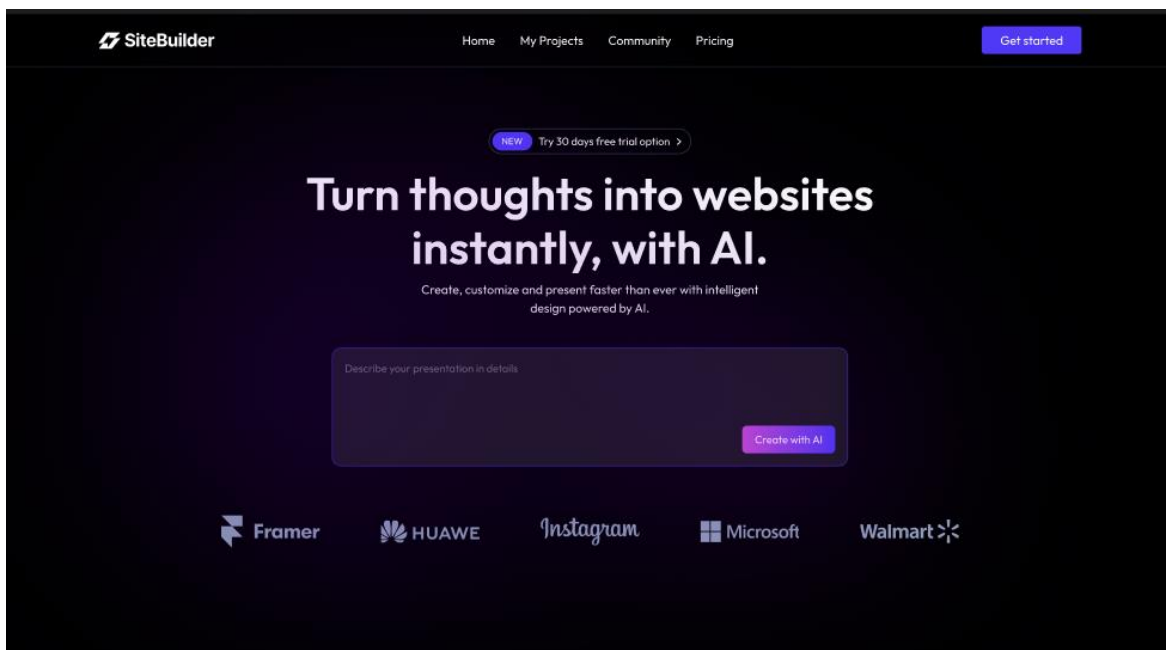
Figure 5.1: Ai website builder workflow

System Architecture Overview

The AI Powered Website Builder follows a three-tier architecture consisting of:

1. **Frontend Layer (React.js)**
2. **Backend Layer (Node.js & Express.js)**
3. **Database & AI Integration Layer (Neon PostgreSQL + OpenRouter API)**

The architecture enables smooth communication between the user interface, backend services, AI processing, and cloud database storage.





Frontend Layer (Client Side)

The frontend is developed using **React.js** and provides an interactive user interface for users to generate websites using prompts.

Functions of Frontend

- User Registration & Login
- Prompt Input Form
- Website Preview Interface
- Generated Code Display
- Project Dashboard
- Website Editing Interface

Workflow

1. User enters website requirements such as:
 - Website type
 - Theme
 - Features
 - Color preferences
2. React frontend sends the prompt request to backend APIs.
3. Generated AI response is displayed dynamically.
4. Users can edit and save generated websites.

Authentication Module

The authentication system is implemented using Better Auth to provide secure and modern user authentication features. Better Auth simplifies authentication management by handling sessions, login flows, and security mechanisms efficiently.

Features of Better Auth

- Secure user registration and login
- Session-based authentication
- Email and password authentication
- Protected routes and APIs
- Secure cookie management
- Scalable authentication architecture

Working Process

1. User registers using email and password.
2. Better Auth validates user credentials.
3. Session is created securely after successful login.
4. Authentication cookies/session tokens are managed automatically.
5. Protected routes allow access only to authenticated users.

Advantages

- More secure session handling
- Simplified authentication integration
- Easy backend configuration
- Better developer experience
- Improved scalability and maintainability

OpenRouter API Integration

The OpenRouter API is integrated to provide AI-generated website content and layouts.

Working Process

1. Backend sends prompt to OpenRouter API
2. AI model processes request
3. Response is returned as generated content/code
4. Backend formats response
5. Frontend displays website preview

Complete Workflow of System

Step-by-Step Working

Step 1: User Login

- User logs into system using credentials.

**Step 2: Prompt Submission**

- User enters website requirements.

Step 3: Frontend Request

- React frontend sends prompt to backend API.

Step 4: Backend Processing

- Backend validates request.
- Creates API request for OpenRouter.

Step 5: AI Generation

- OpenRouter AI generates:
 - Website content
 - Structure
 - Layout ideas
 - Code snippets

Step 6: Response Handling

- Backend receives AI response.
- Processes generated output.

Step 7: Database Storage

- Generated project is stored in Neon database.

Step 8: Website Preview

- Frontend displays generated website.
- User can edit or save project.

VI. TECHNOLOGIES USED

- **Frontend:** React.js
- **Backend:** Node.js, Express.js
- **Database:** PostgreSQL (Neon DB)
- **AI API:** OpenRouter API
- **Authentication:** Better Auth
- **Version Control:** Git & GitHub
- **Deployment:** Vercel / Hostinger

VII. RESULT

- The system successfully generates website content using AI.
- Users can create website layouts through prompts.
- Generated projects are stored securely in Neon database.
- The frontend interface is responsive and user-friendly.
- AI-generated content reduces development time significantly.
- Multiple website templates can be generated dynamically.
- Real-time preview functionality improves user experience.

VII. CONCLUSION

The AI Powered Website Builder using PERN Stack, Neon Database, and OpenRouter API provides an efficient and intelligent solution for automated website creation. The system successfully integrates AI capabilities with modern web technologies to generate website content and layouts dynamically. By reducing manual effort and simplifying development processes, the project enables users to create professional websites easily and quickly. The cloud-based database and scalable architecture make the system reliable, flexible, and suitable for future enhancements.



REFERENCES

- [1]. J. Smith and A. Brown, “**AI Powered Website Generation Using Large Language Models,**” International Journal of Computer Science and Engineering, Vol. 11, Issue 4, pp. 210–218, 2023.
- [2]. M. Johnson, R. Lee, and P. Kumar, “**Automated Web Development Using Artificial Intelligence,**” IEEE Xplore, 2022.
- [3]. S. Patel and D. Sharma, “**Smart Website Builder Using MERN/PERN Stack and AI Integration,**” International Journal of Scientific Research in Computer Science, Vol. 10, Issue 6, pp. 55–62, 2023.
- [4]. A. Gupta and R. Verma, “**AI Based Dynamic Website Generation System,**” International Journal of Advanced Research in Computer Science, Vol. 13, Issue 3, pp. 120–127, 2022.
- [5]. P. Roy and K. Mehta, “**Cloud Based Website Builder Using React and Node.js,**” International Journal of Engineering Research & Technology (IJERT), Vol. 9, Issue 8, 2021.
- [6]. T. Wilson and J. Clark, “**Integration of AI Models in Full Stack Web Applications,**” researchgate.net/publication/AI_Web_Application_Integration_Systems, DOI:10.1088/1757-899X/2023/042075.
- [7]. R. Singh, M. Tiwari, and A. Deshmukh, “**AI Driven Website Generation Using OpenRouter API,**” International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 12, Issue 5, pp. 2245–2252, 2024.
- [8]. M. Hasan, S. Hossain, and A. Alamri, “**Smart Web Platforms Using Artificial Intelligence and Cloud Databases,**” IEEE Internet of Things and Cloud Computing Journal, Vol. 8, Issue 4, pp. 5237–5245, 2023.
- [9]. T. Gunasundari and K. Priya, “**Responsive AI Website Builder for Modern Web Applications,**” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 7, Issue 2, pp. 276–281, 2022.
- [10]. B. Rao and L. Rao, “**Website Automation Using React.js, Node.js, and PostgreSQL,**” International Journal of Advanced Trends in Computer Science and Engineering, Vol. 10, Issue 5, pp. 7642–7647, 2021.
- [11]. D. Sharma, P. Gupta, and R. Kumar, “**Artificial Intelligence Based Smart Website Builder,**” International Journal of Engineering Research & Technology (IJERT), Vol. 11, Issue 7, pp. 133–140, 2023.
- [12]. P. Soujanya and B. Praveen, “**AI Assisted Website Generation System Using Cloud Technologies,**” International Journal of Engineering Research & Technology (IJERT), 2022.