CHAPTER III: SYSTEM DESIGN

3.1. System Architecture and Overview

System design is the process of defining the architecture components, modules, interface and data for a system to satisfy specified requirements. It involves specifying how the system will accomplish its objectives and meet the needs of users, considering factors such as efficiency, scalability and maintainability. It is a crucial phase in the software development life cycle (SDLC) and involves transforming the requirements gathered during the analysis phase into a blueprint for the actual system. We have developed "CONTENT MANAGEMENT SYSTEM (GYANSHRISTI)" which required MongoDB for its functioning. To use this system, we need a web browser, internet, laptop/ desktop and the users.

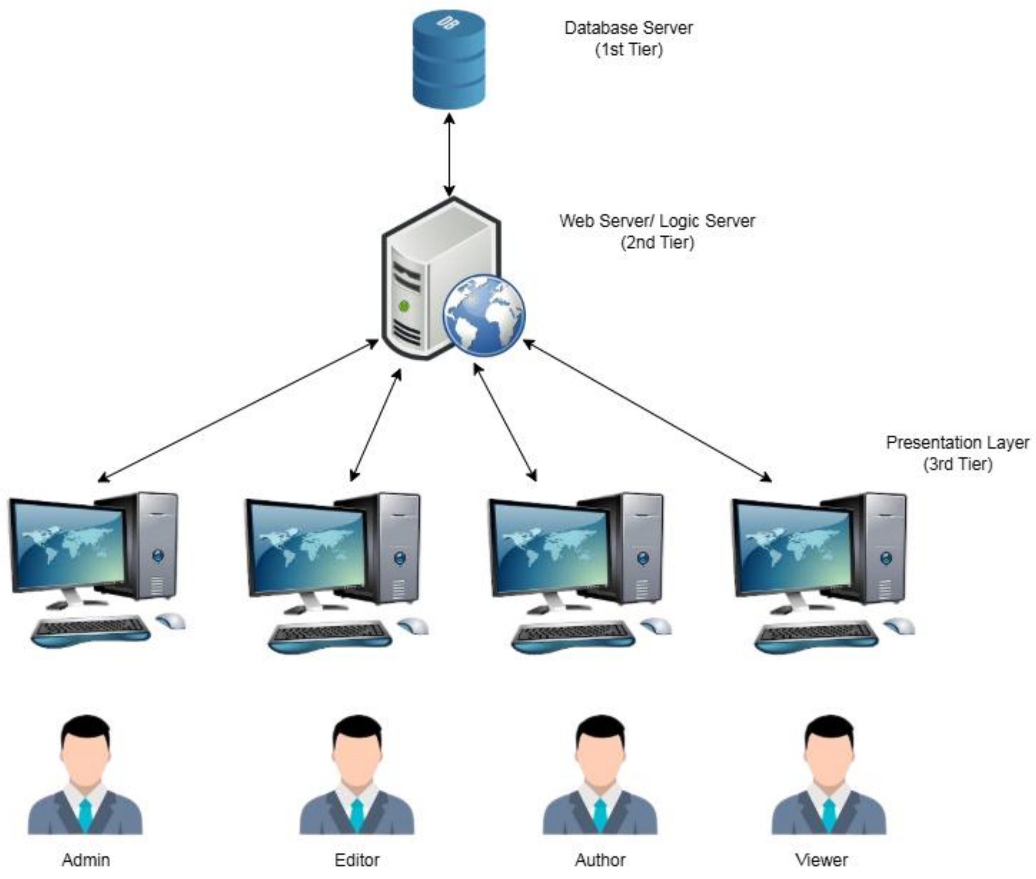


Figure 3.1: System Architecture

3.2. System Design

System design is the process of planning and defining the structure, components, interfaces, and data of a system to fulfill specified requirements. It involves creating a detailed blueprint that guides the implementation of the system, covering aspects such as architecture, components, data organization, interfaces, procedures, and security.

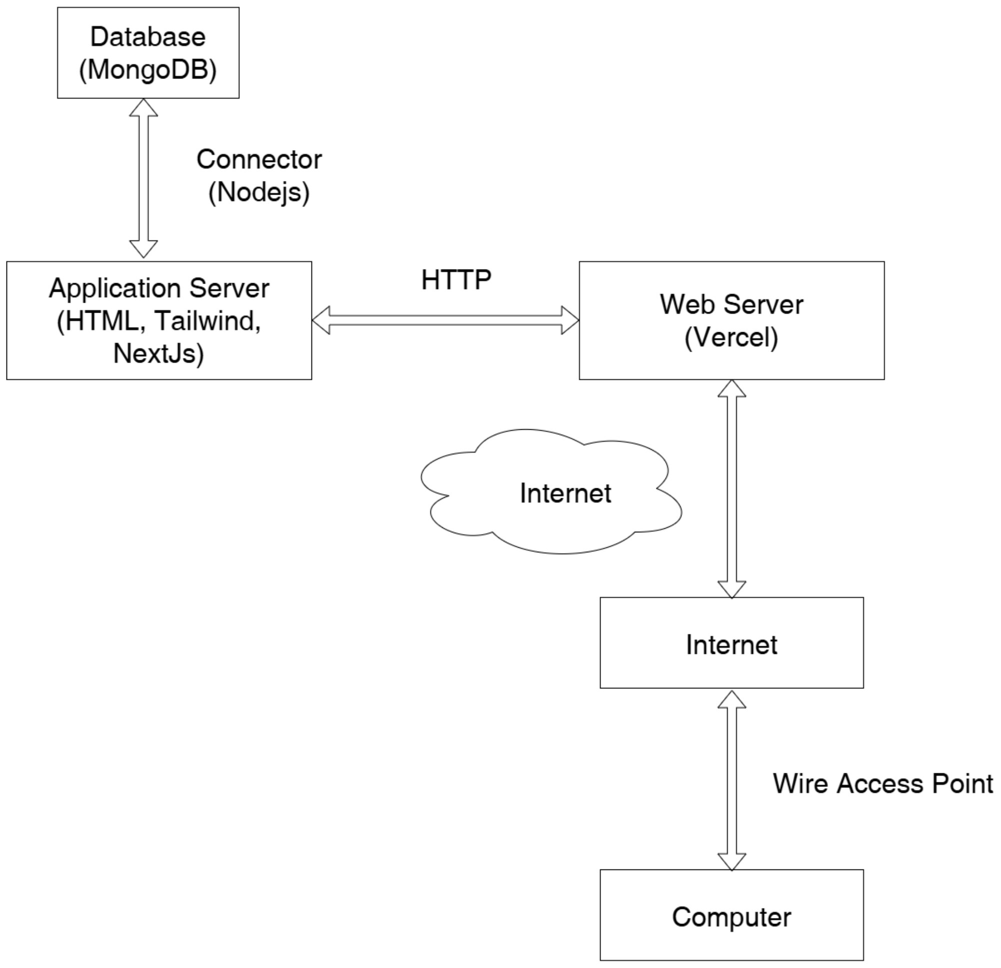
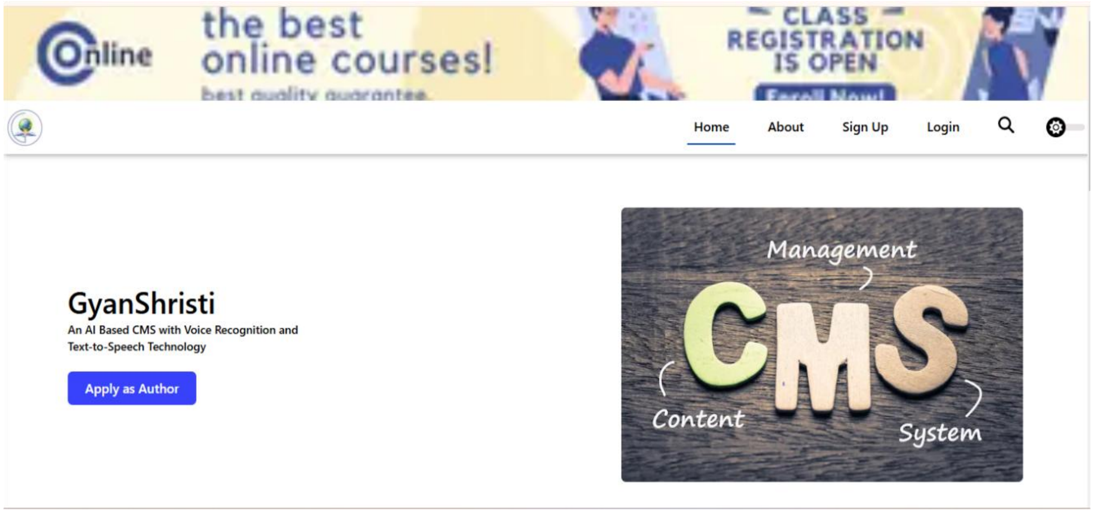


Figure 3.2: Implementation architecture of Content Management System

3.3. Interface Design

Interface design is the process of creating visual and interactive elements for computer systems, software, websites, or applications to make them user-friendly and aesthetically pleasing. It involves designing the layout, appearance, and functionality of on-screen elements such as buttons, menus, and navigation, with the aim of providing a positive and efficient user experience.

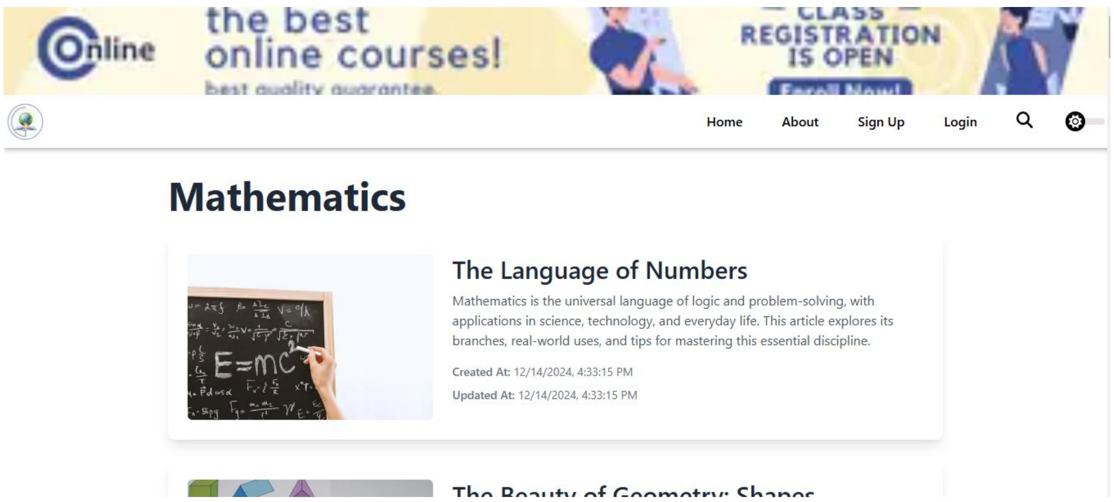


GyanShristi

An AI Based CMS with Voice Recognition and Text-to-Speech Technology

Apply as Author

Figure 3.3: Homepage of Viewer



Mathematics

The Language of Numbers

Mathematics is the universal language of logic and problem-solving, with applications in science, technology, and everyday life. This article explores its branches, real-world uses, and tips for mastering this essential discipline.

Created At: 12/14/2024, 4:33:15 PM

Updated At: 12/14/2024, 4:33:15 PM

The Beauty of Geometry: Shapes

Figure 3.4: Category Wise Content of Viewer

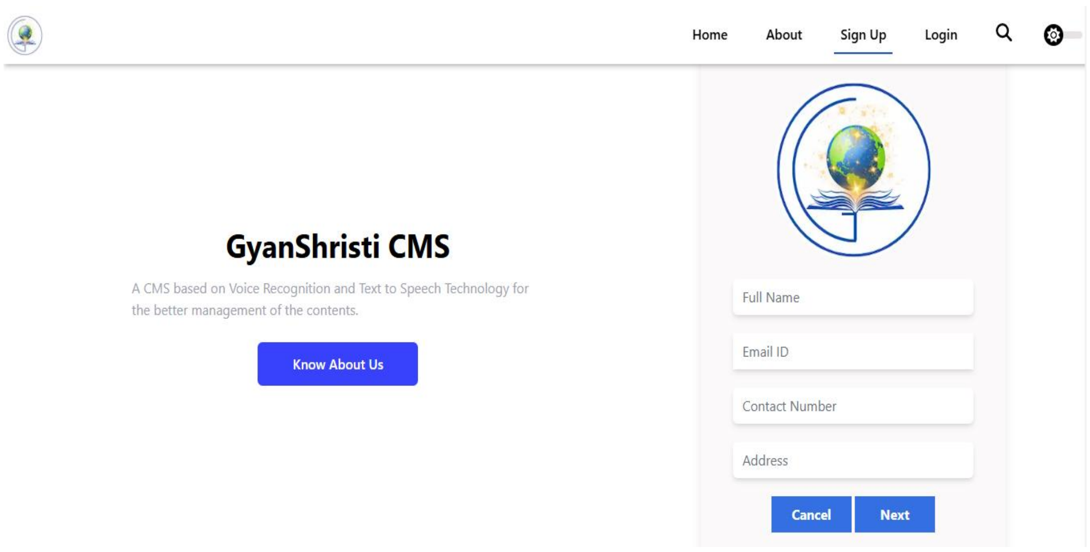

Home
About
Sign Up
Login
GyanShristi CMS
A CMS based on Voice Recognition and Text to Speech Technology for the better management of the contents.
Know About Us
Full Name
Email ID
Contact Number
Address
Cancel
Next

Figure 3.5: Sign up Page of Non-registered User

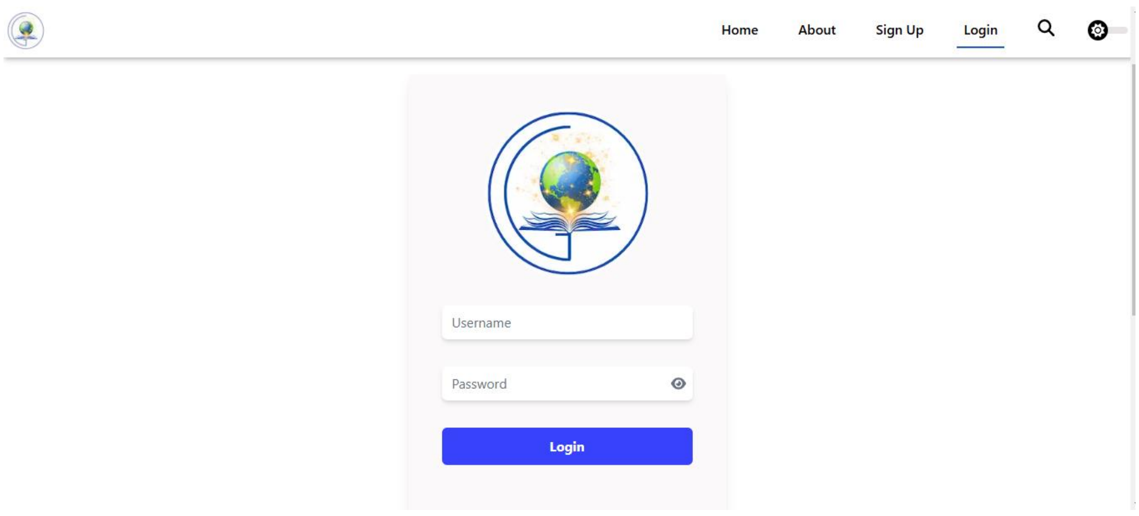


Figure 3.6: Login Page

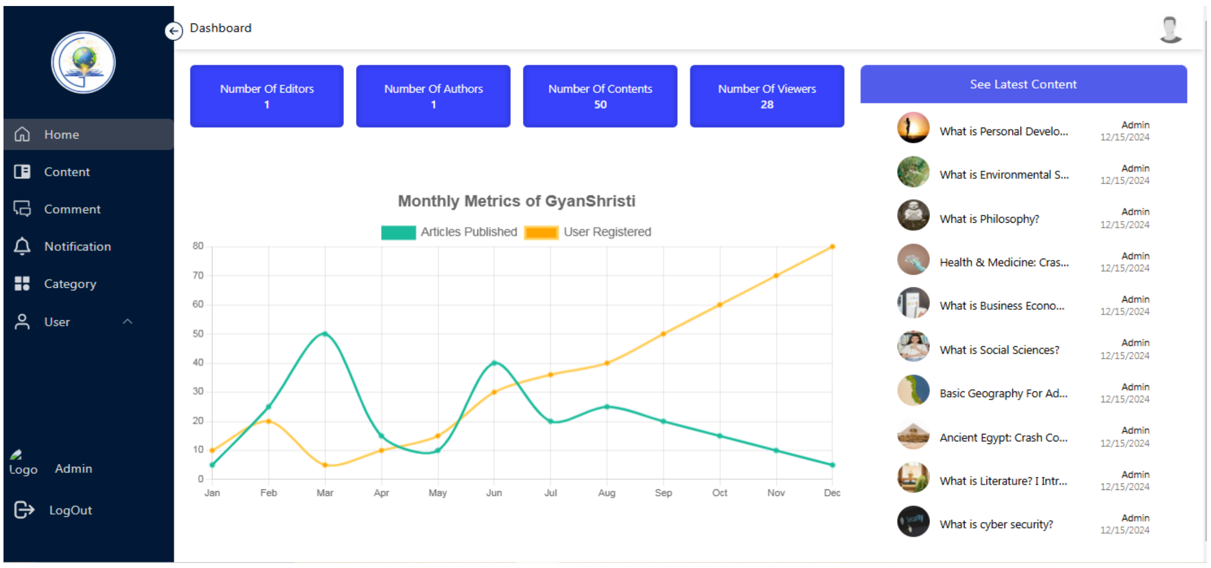

Dashboard
Home
Content
Comment
Notification
Category
User
Logo
Admin
LogOut
Number Of Editors 1
Number Of Authors 1
Number Of Contents 50
Number Of Viewers 28
Monthly Metrics of GyanShristi
Articles Published
User Registered
See Latest Content
What is Personal Develo... Admin 12/15/2024
What is Environmental S... Admin 12/15/2024
What is Philosophy? Admin 12/15/2024
Health & Medicine: Cras... Admin 12/15/2024
What is Business Econo... Admin 12/15/2024
What is Social Sciences? Admin 12/15/2024
Basic Geography For Ad... Admin 12/15/2024
Ancient Egypt: Crash Co... Admin 12/15/2024
What is Literature? I Intr... Admin 12/15/2024
What is cyber security? Admin 12/15/2024

Figure 3.7: Dashboard Page of Admin

3.4. Database Schema

A database schema in MongoDB defines the structure of data in collections and documents, allowing flexibility with varying fields, while tools like Mongoose can enforce consistent patterns and validation.

3.4.1 UserSchema

```
const UserSchema = new mongoose.Schema({
name: {
type: String,
required: true,
},
username: {
type: String,
required: true,
},
password: {
type: String,
required: true,
},
email: {
type: String,
required: true,
},
address: {
type : String,
required : true,
},
profile_pic: {
type: String,
},
phone_number: {
type: Number,
required: true,
},
role: {
type: String,
enum: ["admin", "editor", "author", "viewer"],
default: "viewer",
},
status: {
type: String,
enum: ["unrequested", "pending", "approved","rejected"],
},
otp:{
type: String
},
created_at: {
type: Date,
default: Date.now,
},
});
```

3.4.2 ContentSchema

```
const ContentSchema = new mongoose.Schema({
title: {
type: String,
required: true,
},
description: {
type: String,
required: true,
},
content_type: {
type: String,
enum: ["pdf", "video", "post"],
required: true,
},
location: {
type: String,
},
blog: {
type: String,
},
thumbnail: {
type: String,
default:
"https://images.pexels.com/photos/1323550/pexels-photo-
1323550.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1",
},
status: {
type: String,
enum: ["Pending", "Uploaded", "Rejected"],
default: "Pending",
},
user_id: {
type: mongoose.Schema.Types.ObjectId,
ref: "User",
},
category_id: {
type: mongoose.Schema.Types.ObjectId,
ref: "Category",
default: null,
},
created_at: {
type: Date,
default: Date.now,
},
updated_at: {
type: Date,
default: Date.now,
},
});
```

3.4.3 CommentSchema

```
const CommentSchema = new mongoose.Schema({
description : {
type : String,
required : true
},
user_id : {
type : mongoose.Schema.Types.ObjectId,
ref : 'User'
},
content_id : {
type : mongoose.Schema.Types.ObjectId,
ref : 'Content'
},
parent_comment_id :[{
type : mongoose.Schema.Types.ObjectId,
ref : 'Comment'
}],
created_at : {
type : Date,
default : Date.now
}
});
```

3.4.4 CategorySchema

```
const CategorySchema = new mongoose.Schema({
title: {
type: String,
required: true,
},
user_id: {
type: mongoose.Schema.Types.ObjectId,
ref : "User"
}
});
```

3.4.5 LoginSchema

```
const LoginSchema = new mongoose.Schema({
user_id: {
type: mongoose.Schema.Types.ObjectId,
required: true,
},
created_at: {
type: Date,
default: Date.now,
},
});
```