



# Final System Design – **CodingQuestions.in**

---



## 1. Project Title

**CodingQuestions.in – A Simple Q&A and Coding Knowledge Sharing Platform**

---



## 2. Problem Statement

Beginners and students often lack a clear, centralized platform to ask doubts and learn coding topics in an organized way. Most existing resources are either unstructured, too complex, or fragmented across different websites.

---



## 3. Introduction

The rapid growth of technology and digital learning platforms has created an immense demand for structured and user-friendly systems where students and beginners can learn programming effectively. While there are many online resources available, most are either scattered across different websites, unorganized, or too complex for newcomers. This makes it difficult for learners to clarify doubts, practice coding concepts, and engage in meaningful discussions.

The proposed system, **CodingQuestions.in**, is designed to solve these challenges by providing a centralized Q&A platform dedicated to coding and programming knowledge sharing. It allows users to ask questions, provide answers, explore learning materials, and interact with peers and mentors in an organized manner. With separate modules for users, administrators, and categorized tutorials, the platform ensures a smooth learning experience, easy navigation, and effective management of content.

This system not only simplifies the process of doubt-solving but also encourages collaborative learning, making coding education more accessible, engaging, and efficient for students.

---



## 4. Objectives

- Allow users to register/login
  - Let users ask and answer questions
  - Organize questions by programming category
  - Provide learning/tutorial content per category
  - Admin can manage users, questions, and categories
- 

## 5. Existing System

Currently, beginners and students rely on multiple online resources such as forums, coding websites, and social media groups to clarify doubts and learn programming concepts. These platforms are often unorganized and lack a structured way of categorizing questions and answers.

Students usually:

- Search answers on scattered websites (e.g., StackOverflow, Quora, blogs).
- Learn from tutorials available on YouTube or other coding websites.
- Depend on classroom discussions or peers for help.

### Drawbacks of Existing System

- Information is fragmented across multiple platforms.
  - Beginners face difficulty identifying accurate and verified answers.
  - No proper categorization of programming languages and topics.
  - Lack of real-time or interactive learning features.
  - Limited or no authentication/security for users.
  - No dedicated community-driven support exclusively for coding beginners.
- 

## 6. Proposed System

(A Q&A Platform with Learning Features)

## Introduction

- A platform where users can ask and answer questions
- Supports learning programming languages
- Provides secure login and signup options

## Key Features (Authentication)

- User Login & Signup
- Email verification with OTP on mobile
- Login with OTP
- Google OAuth Login

## Key Features (User Actions)

- Users can ask questions
- Users can answer questions
- Users can update their questions and answers
- Users can learn programming languages
- Users can answer other users' questions

## Benefits of the System

- Secure login with multiple authentication methods
- Interactive learning through Q&A
- Community-driven knowledge sharing
- Easy to use for beginners and programmers

## Conclusion

- The proposed system provides a secure and collaborative platform
  - Encourages learning and sharing among users
  - Enhances programming knowledge in a simple and effective way
- 

## 7. System Users

The system consists of two primary types of users: **End Users** (students, learners, programmers) and **Administrators** (platform managers). Each user group has specific roles and responsibilities.

### 1. End Users (Students / Learners / Programmers)

#### **Roles:**

- Register and create an account (manual signup, OTP login, or Google OAuth)
- Post questions related to coding and programming concepts
- Provide answers, solutions, and suggestions to other users' questions
- Edit or update their own questions and answers
- Browse questions by categories such as Java, Python, C++, etc.
- Learn through tutorials and reference materials available in the platform
- Interact with peers and share coding knowledge in a collaborative way

### 2. Administrators (Backend Access Only)

#### **Roles:**

- Manage all registered users (view, edit, or delete accounts if necessary)
- Oversee posted questions and answers to maintain content quality
- Remove irrelevant, duplicate, or spam content
- Create, edit, or delete categories for better organization
- Ensure system security and enforce platform policies

- Handle user reports and feedback for continuous improvement
- 



## 8. Core Functional Modules



### User Panel

- User Registration (manual or Google)
- Login/Logout with session management
- Ask a question (title + description + category)
- Answer/comment on existing questions
- View questions and answers by category



### Q&A System

- Ask question → stored in thread
- Answer/comment → stored in comment
- Questions grouped by category
- Questions display username, content, and timestamp



### Learning / Tutorial Section (optional, extendable)

- Categories like Java, Python, C++
- Each category has its description and can link to learning pages
- Extendable with tutorials table if needed



### Admin Panel (Backend Access Only)

- View/delete users
- Edit/delete questions (thread)

- Edit/delete comments (comment)
  - Manage categories
- 

## 9. Technology Stack

The system is built using a combination of modern web technologies that ensure scalability, usability, and security.

Layer	Technology Used
Frontend	HTML5, CSS3, JavaScript, Tailwind CSS
Backend	Core PHP
Database	MySQL
Authentication	PHP Sessions + Google OAuth (optional)
Hosting	cPanel or Shared Hosting
Tools	VS Code, phpMyAdmin, GitHub, XAMPP

---

## 10. Core Database Tables

- **users** → Stores user info, password, login method
  - **thread** → Stores questions (title, description, user, category)
  - **comment** → Stores answers/comments for questions
  - **category** → Stores programming categories (Java, C++, etc.)
  - **password\_resets** → Stores OTPs for password reset
- 



## 11. ER Diagram Structure (Simplified)

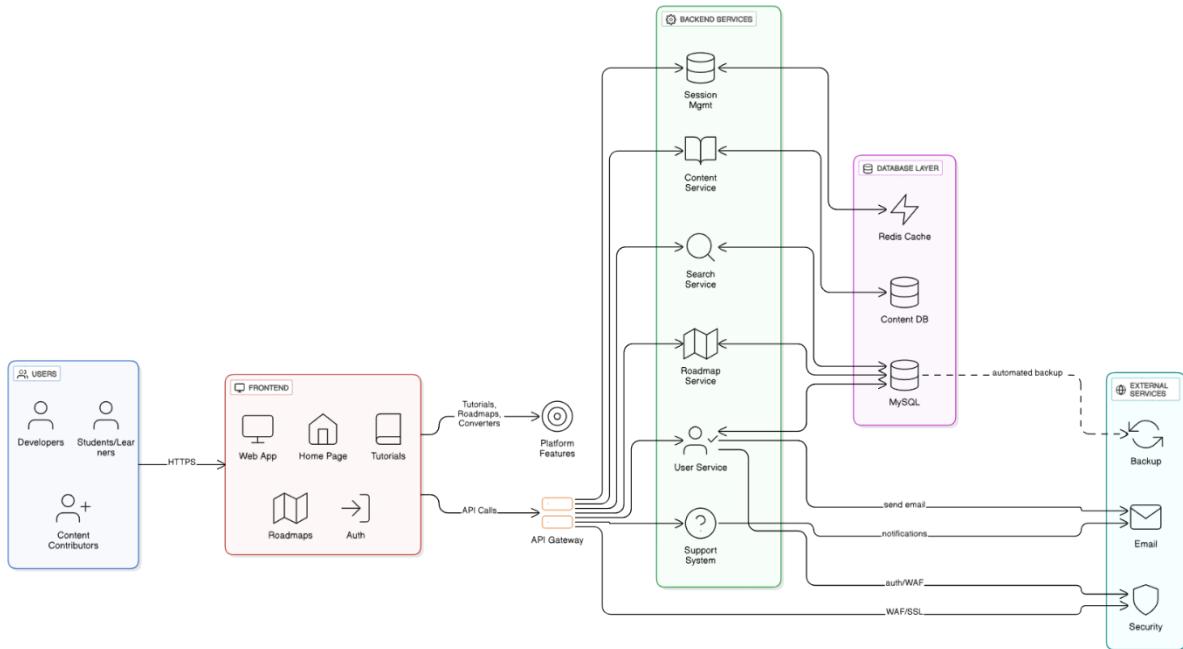
- **users**
- |---< **thread** (**questions**)
- |---< **comment** (**answers**)
- ---> **category**



## 12. Security Considerations

- Passwords hashed with bcrypt
- Input sanitization (basic XSS prevention)
- CAPTCHA recommended for forms
- OTP-based password recovery in place
- Session-based login with timeout suggested

# System Architecture Diagram



# Database Development

---

## Tables Overview

The database consists of 5 tables:

- category
  - comment
  - password\_resets
  - thread
  - users
- 

### 1. category

Purpose:

Stores programming categories with descriptions.

Structure:

- cate\_no (INT, Primary Key, Auto Increment): Unique ID for each category.
  - cate\_name (VARCHAR(100)): Category name (e.g., C, Python).
  - cate\_des (TEXT): Description of the category.
- 

### 2. comment

Purpose:

Stores user comments linked to threads.

Structure:

- comm\_id (INT, Primary Key, Auto Increment): Unique ID for each comment.
- comm\_con (TEXT): Comment content.
- thr\_no (INT, Foreign Key): Linked to thread.thr\_id.
- user\_id (BIGINT, Foreign Key): Linked to users.user\_id.
- created\_at (TIMESTAMP): Auto-stamped comment creation date.

Relationships:

- `thr_no` → `thread(thr_id)` (ON DELETE CASCADE)
  - `user_id` → `users(user_id)`
- 

### 3. password\_resets

Purpose:

Manages OTP-based password reset functionality.

Structure:

- `id` (INT, Primary Key, Auto Increment)
  - `email` (VARCHAR(255)): User email for password reset
  - `otp` (VARCHAR(6)): One-Time Password
  - `expiry_time` (DATETIME): OTP expiry
  - `created_at` (TIMESTAMP): Time of OTP generation
- 

### 4. thread

Purpose:

Manages discussion threads.

Structure:

- `thr_id` (INT, Primary Key, Auto Increment): Unique ID for thread.
- `thr_title` (VARCHAR(255)): Thread title.
- `thr_dec` (TEXT): Thread description.
- `thr_cat_id` (INT, Foreign Key): Linked to `category.cate_no`.
- `thr_user_id` (INT, Foreign Key): Linked to `users.user_id`.
- `created_at` (TIMESTAMP): Thread creation time.

Relationships:

- `thr_cat_id` → `category(cate_no)`
  - `thr_user_id` → `users(user_id)`
- 

### 5. users

Purpose:

Stores registered user details.

## Structure:

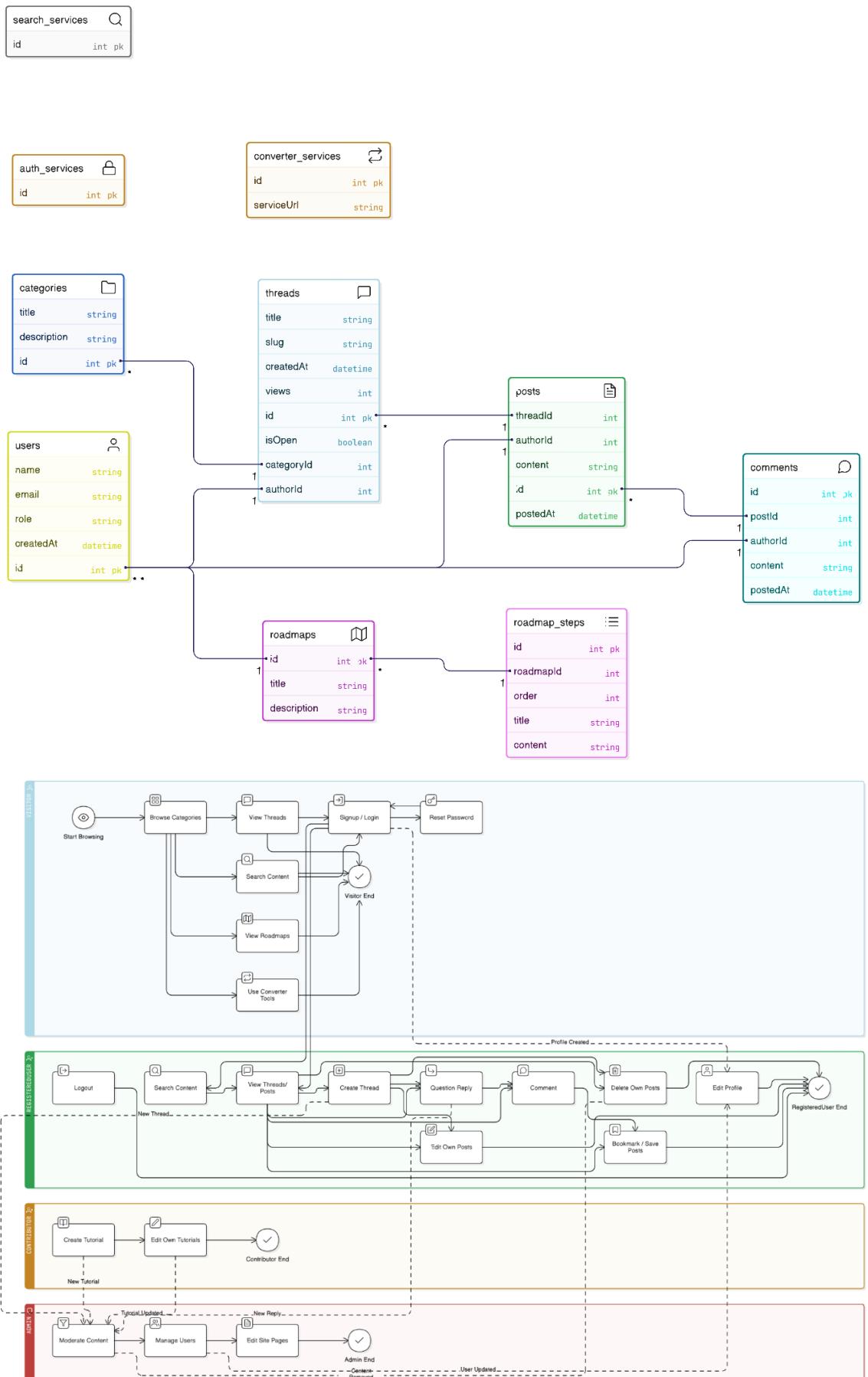
- `user_id` (INT, Primary Key, Auto Increment)
  - `username` (VARCHAR(100))
  - `email` (VARCHAR(150), Unique)
  - `password` (VARCHAR(255)): Hashed password
  - `google_id` (VARCHAR(255)): If logged in via Google
  - `auth_provider` (ENUM: 'manual', 'google'): Authentication type
  - `otp` (VARCHAR(6)): OTP for email verification
  - `otp_expiry` (DATETIME): OTP expiry
  - `created_at` (TIMESTAMP)
- 

## Relationships (ERD Concept)

- `users` → (1-to-many) → `thread` → (1-to-many)  
→ `comment`
  - `category` → (1-to-many) → `thread`
  - `password_resets` is standalone but linked logically to `users.email`
- 

## Key Features

-  OTP-based authentication and password reset
-  Category-wise thread organization
-  User comments linked to threads
-  Supports Google login via `google_id`
-  Cascading deletes maintain referential integrity



# CodingQuestions.in - Data Dictionary

## 1. Table: category

Field	Data Type	Size	Constraints	Description
cate_no	INT	11	PK, AUTO_INCREMENT	Unique ID for each category.
cate_name	VARCHAR	100	NOT NULL	Name of the category.
cate_des	TEXT	-	NOT NULL	Description of the category.

## 2. Table: comment

Field	Data Type	Size	Constraints	Description
comm_id	INT	11	PK, AUTO_INCREMENT	Unique ID for each comment.
comm_con	TEXT	-	NOT NULL	Content of the comment.
thr_no	INT	11	FK (thread.thr_id)	Thread ID linked.
user_id	BIGINT	20	FK (users.user_id)	User ID of comment creator.
created_at	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP	Comment creation time.

## 3. Table: password\_resets

Field	Data Type	Size	Constraints	Description
id	INT	11	PK, AUTO_INCREMENT	Unique ID for reset request.
email	VARCHAR	255	NOT NULL	Email of the user.
otp	VARCHAR	6	NOT NULL	One-time password.
expiry_time	DATETIME	-	NOT NULL	OTP expiry date.
created_at	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP	Reset request time.

## 4. Table: thread

Field	Data Type	Size	Constraints	Description
thr_id	INT	11	PK, AUTO_INCREMENT	Unique ID for thread.
thr_title	VARCHAR	255	NOT NULL	Thread title.
thr_dec	TEXT	-	NOT NULL	Thread description.
thr_cat_id	INT	11	FK (category.cate_no)	Linked category ID.
thr_user_id	INT	11	FK (users.user_id)	User ID of creator.
created_at	TIMESTAMP	-	DEFAULT CURRENT_TIMESTAMP	Thread creation time.

## 5. Table: users

Field	Data Type	Size	Constraints	Description
user_id	INT	11	PK, AUTO_INCREMENT	Unique ID for user.
username	VARCHAR	100	NULLABLE	Username of user.
email	VARCHAR	150	UNIQUE, NOT NULL	User email ID.
password	VARCHAR	255	NULLABLE	Hashed password.
google_id	VARCHAR	255	NULLABLE	Google OAuth ID.
auth_provider	ENUM	-	DEFAULT 'manual'	Auth method.
otp	VARCHAR	6	NULLABLE	OTP for verification.

## CodingQuestions.in - Data Dictionary

### Relationships

- A user can create multiple threads (users to thread via thr\_user\_id).
- A thread can have multiple comments (thread to comment via thr\_no).
- A category can contain multiple threads (category to thread via thr\_cat\_id).
- The password\_resets table is logically linked to users via the email field.

The screenshot displays the Codingque website interface across multiple sections:

- Header:** Features the Codingque logo, a search bar, and navigation links for Home, Support, How to Learn, Converters, Tutorials, and Roadmaps.
- Search Bar:** A central search bar with a placeholder "Search..." and a "Search" button.
- Language Pages:**
  - C:** Describes C as a powerful general-purpose programming language used in system and application development. Includes a "View Threads" button.
  - C++:** Describes C++ as builds on C and adds object-oriented features, commonly used in game development. Includes a "View Threads" button.
  - Java:** Describes Java as a versatile, object-oriented language used in enterprise applications, Android dev., etc. Includes a "View Threads" button.
  - Python:** Describes Python as an easy-to-read, high-level language great for web development, data science, ml..., etc. Includes a "View Threads" button.
  - PHP:** Describes PHP as a server-side scripting language especially suited for web development. It powers m... Includes a "View Threads" button.
  - JavaScript:** Describes JavaScript as a dynamic, client-side language used to create interactive web pages, often... Includes a "View Threads" button.
- Content Area:** Displays a "C++" section with a brief description, a warning box about inappropriate language, and a "To ask a question, please sign up or log in to your account." message.
- Questions Section:** Shows a question posted by "Manthan" on July 6, 2025, at 08:38:09. The question text is "Hello test".
- Footer:** A dark sidebar containing links to About Codingque, Resources (C Language, C++ Language, PHP, HTML, Javascript, API Integration), Roadmaps (Frontend, Backend, Full Stack, Java, Javascript, React.js, C++, Python, Node.js), Account (Create Account, Privacy Policy, User Agreement), Contact Us (Contact, Privacy Policy, User Agreement), and a "Visit Interview Questions" link.
- Page Bottom:** Copyright notice: "© 2024 Codingque. All rights reserved."

## Create Account

Username

Email

Password

Confirm Password

**Sign Up**

 Sign Up with Google

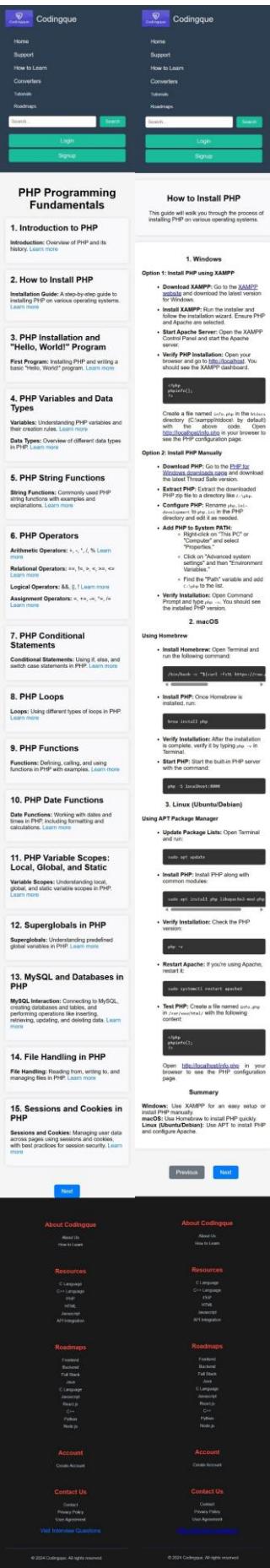
# User Login

Email

Password

**Login**

[Forgot your password?](#)



PHP Programming Fundamentals

1. Introduction to PHP

2. How to Install PHP

3. PHP Installation and "Hello, World!" Program

4. PHP Variables and Data Types

5. PHP String Functions

6. PHP Operators

7. PHP Conditional Statements

8. PHP Loops

9. PHP Functions

10. PHP Date Functions

11. PHP Variable Scopes: Local, Global, and Static

12. Superglobals in PHP

13. MySQL and Databases in PHP

14. File Handling in PHP

15. Sessions and Cookies in PHP

Using XAMPP

Using PHP Manager

Using Homebrew

Using APT Package Manager

Windows: Use XAMPP as an easy setup or macOS: Use Homebrew to install PHP quickly

Linux (Ubuntu/Debian): Use APT to install PHP and configure Apache

Summary

Previous Next

### Two-Step Verification

Please enter the 6-digit OTP code sent to your email: manthanpatel26510@gmail.com

Enter OTP Code:

**Verify OTP**

Codingque

Home Support How to Learn Converters Tutorials Roadmaps Search... Search Login Signup

# Hello

test

**Warning:** Be respectful. Avoid spam, NSFW content, and offensive language.

Posted by: *Manthan*

## Answers

 Comment by: *Manthan Patel*  
Posted: 2025-07-06 10:32:36  
how are you

 Comment by: *Manthan Patel*  
Posted: 2025-07-06 10:23:46  
how are you

 Comment by: *Manthan Patel*  
Posted: 2025-07-06 10:22:38  
who are you

Please [login](#) or [signup](#) to post an answer.

---

**About Codingque**

- [About Us](#)
- [How to Learn](#)

**Resources**

- [C Language](#)
- [C++ Language](#)
- [PHP](#)
- [HTML](#)
- [Javascript](#)
- [API Integration](#)

**Roadmaps**

- [Frontend](#)
- [Backend](#)
- [Full Stack](#)
- [Java](#)
- [C Language](#)
- [Javascript](#)
- [React.js](#)
- [C++](#)
- [Python](#)
- [Node.js](#)

**Account**

- [Create Account](#)

**Contact Us**

- [Contact](#)
- [Privacy Policy](#)
- [User Agreement](#)

[Visit Interview Questions](#)

© 2024 Codingque. All rights reserved.

The screenshot shows the Codingque website interface. At the top is a dark blue header with the Codingque logo and a search bar. Below the header is a sidebar with links: Home, Support, How to Learn, Converters, Tutorials, and Roadmaps. There's also a search bar and a "Logout" button. A teal banner at the bottom of the sidebar says "Your Q&A". The main content area has a "Welcome, Manthan" message. Below it is a section titled "Your Questions and Answers" with a sub-section "Your Questions" containing a single item: "Title: Hello" and "Description: test" with an "Edit" button. The "Your Answers" section below it shows "No answers found for this user." On the left side of the main content area is a sidebar with sections: "About Codingque" (About Us, How to Learn), "Resources" (C Language, C++ Language, PHP, HTML, Javascript, API Integration), "Roadmaps" (Frontend, Backend, Full Stack, Java, C Language, Javascript, React.js, C++, Python, Node.js), "Account" (Create Account), and "Contact Us" (Contact, Privacy Policy, User Agreement). A link "Visit Interview Question" is also present.

# Submit Your Problem

Title

Please enter a short problem title.

Description

**Submit**

+

# Submit Your Answer

Write your answer  
here...

Your Answer

Post Answer

← Back to Thread