Design/Architecture Document

1. Introduction

1.1 Purpose

This document provides a detailed description of the system architecture and design for the task management tool.

1.2 Scope

The design covers the frontend, backend, and database components of the system, including OOAD/UML diagrams.

2. System Architecture

2.1 Overview

The system follows a three-tier architecture:

- **Presentation Layer (Frontend)**

- **Business Logic Layer (Backend)**

- **Data Layer (Database)**

2.2 Architectural Diagram

```
+----------------------+

|    Web Browser       |

| (HTML, CSS, JS)      |

+-----------+----------+

          |
```

```
          |
+-----------v-----------+
|    Web Server         |
| (PHP, Apache)         |
+-----------+-----------+
          |
          |
+-----------v-----------+
|    Database Server    |
| (MySQL)               |
+-----------------------+
```

3. Detailed Design

3.1 Frontend Design

3.1.1 User Interface Components
- **Login Page:** Allows users to log in.
- **Task List Page:** Displays the list of tasks.
- **Task Form:** Used for creating and editing tasks.

3.1.2 Technologies Used
- HTML
- CSS
- JavaScript (with a framework like React or Vue.js)

3.2 Backend Design

3.2.1 API Endpoints

- **/api/register:** Handles user registration.
- **/api/login:** Handles user authentication.
- **/api/tasks:** CRUD operations for tasks.

3.2.2 Technologies Used

- PHP (with a framework like Laravel)
- RESTful API principles

3.3 Database Design

3.3.1 Tables

- **Users Table**
  - id (Primary Key)
  - email (Unique)
  - password

- **Tasks Table**
  - id (Primary Key)
  - user_id (Foreign Key)
  - title
  - description
  - category
  - status

3.3.2 Relationships

- One-to-Many relationship between Users and Tasks.

4. OOAD/UML Diagrams

4.1 Use Case Diagram

```
[Use Case Diagram: Task Management System]

  +----------------------+
  |     User             |
  +-----------+----------+
              |
              |
 +------------v------------+
 |  Task Management System  |
 +------------+------------+
              |
              |
  +-----------v----------+
  |   Create Task        |
  +-----------+----------+
              |
              |
  +-----------v----------+
  |   Edit Task          |
  +-----------+----------+
              |
              |
  +-----------v----------+
  |   Delete Task        |
```

```
  +-----------+-----------+
              |
              |
  +-----------v-----------+
  |    View Task          |
  +-----------------------+
```

4.2 Class Diagram

```
[Class Diagram: Task Management System]

+-------------------+
|      User         |
+-------------------+
| -id: int          |
| -email: String    |
| -password: String |
+-------------------+
| +register()       |
| +login()          |
+-------------------+
          |
          |
          |
+-----------v-----------+
|      Task             |
+-----------------------+
```

| -id: int |

| -title: String |

| -description: String |

| -category: String |

| -status: String |

| -user_id: int |

+----------------------+

| +create() |

| +edit() |

| +delete() |

| +view() |

+----------------------+

```

**4.3 Sequence Diagram**

```

[Sequence Diagram: Create Task]

User -> Web Browser: Access Create Task Page

Web Browser -> Web Server: Request Create Task Page

Web Server -> Database: Fetch User Data

Database -> Web Server: Return User Data

Web Server -> Web Browser: Display Create Task Page

User -> Web Browser: Enter Task Details and Submit

Web Browser -> Web Server: Send Task Details

Web Server -> Database: Insert Task into Database

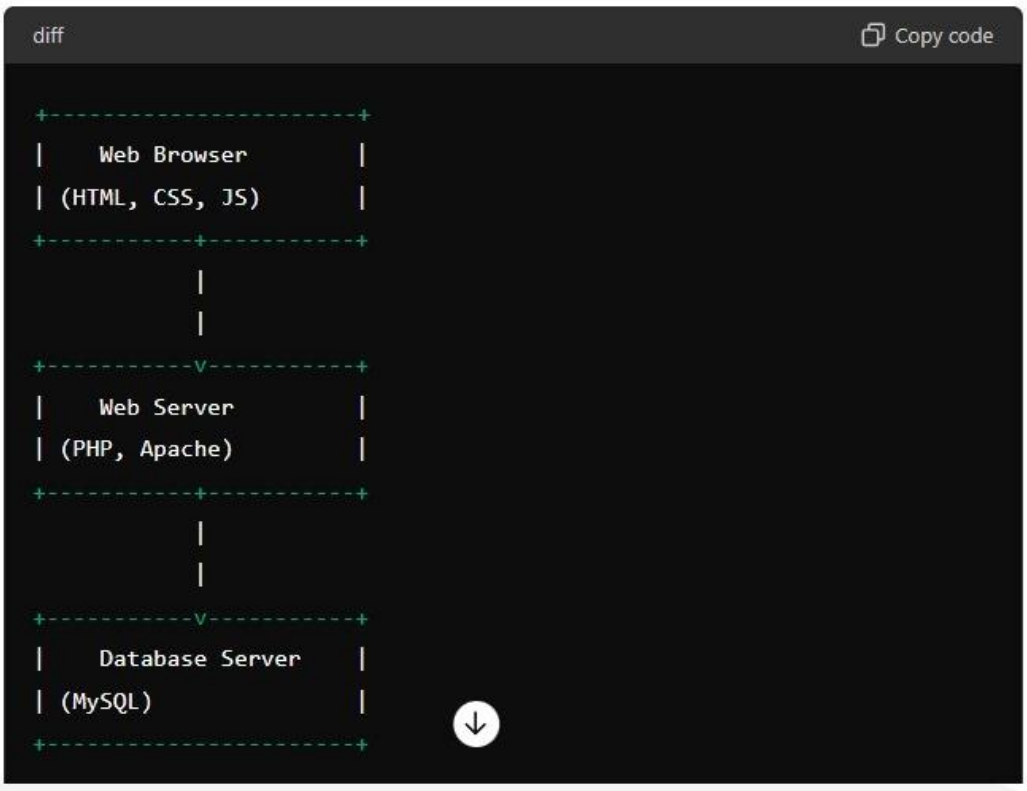
Database -> Web Server: Confirm Task Creation

Web Server -> Web Browser: Display Confirmation

```

## 2.2 Architectural Diagram

diff Copy code


```

4.2 Class Diagram

sql Copy code

```
[Class Diagram: Task Management System]

+-------------------+
|       User        |
+-------------------+
| -id: int          |
| -email: String    |
| -password: String |
+-------------------+
| +register()       |
| +login()          |
+-------------------+
           |
           |
           |
+----------v----------+
|       Task          |
+---------------------+
| -id: int            |
| -title: String      |
| -description: String|
| -category: String   |
| -status: String     |
| -user_id: int       |
+---------------------+
| +create()           |
| +edit()             |
| +delete()           |
| +view()             |
+---------------------+
```

5. Security Considerations

5.1 Authentication and Authorization

- Use JWT (JSON Web Tokens) for session management.

5.2 Data Protection

- Encrypt passwords using bcrypt.

5.3 Input Validation

- Sanitize and validate all user inputs to prevent SQL injection and XSS attacks.

6. Appendix

6.1 Glossary

Definitions of terms and acronyms used in the document.