**Question Paper 1**

1. **(JavaScript, AJAX & PHP)**  
   Create a form where users can enter their name and email. When the form is submitted, use AJAX to send the data to a PHP script that stores it in a database. Display a success message without reloading the page.
2. **(Laravel)**  
   Develop a simple contact management system where users can add, edit, and delete contacts. Store the data in a MySQL database using Laravel.

**Question Paper 2**

1. **(JavaScript, AJAX & PHP)**  
   Create a login form where users enter a username and password. Use AJAX to check credentials in the database and display an error message if they are incorrect—without reloading the page.
2. **(Laravel)**  
   Build a basic student record system where students’ names, roll numbers, and courses can be added, edited, and deleted using Laravel.

**Question Paper 3**

1. **(JavaScript, AJAX & PHP)**  
   Develop a "like" button for blog posts. When clicked, the number of likes should update dynamically using AJAX.
2. **(Laravel)**  
   Create a basic blog system where users can add and delete posts. Show all posts on the homepage.

**Question Paper 4**

1. **(JavaScript, AJAX & PHP)**  
   Build a simple comment section where users can post comments.
2. **(Laravel)**  
   Implement a task manager where users can add tasks with a due date and mark them as completed.

**Question Paper 5**

1. **(JavaScript, AJAX & PHP)**  
   Develop a search box where users can type a name, and matching results from the database appear below dynamically using AJAX.
2. **(Laravel)**  
   Create a simple employee directory where users can add, edit, and delete employee details.

**Question Paper 6**

1. **(JavaScript, AJAX & PHP)**  
   Implement a simple "online users" feature. When a user visits a page, their status updates in the database, and a list of currently online users is displayed.
2. **(Laravel)**  
   Develop a notes application where users can create (with validation) and delete notes. (With every activity a mail should be sent to the admin)

**Question Paper 7**

1. **(JavaScript, AJAX & PHP)**  
   Create a simple to-do list where users can add tasks, and the list updates dynamically without reloading the page.
2. **(Laravel)**  
   Build a Laravel-based book catalog where users can add (with validation) and remove books from a list.

**Question Paper 8**

1. **(JavaScript, AJAX & PHP)**  
   Implement a student attendance system where teachers can mark students as present or absent.
2. **(Laravel)**  
   Develop an image upload system where users can upload and view images.

### ****Question Paper 9****

1. **(JavaScript, AJAX & PHP)**  
   Create a **registration form** where users enter their name and email. When submitted, use AJAX to store the data in a MySQL database and display a success message without reloading the page.
2. **(Laravel)**  
   Develop a **simple product inventory system** where users can add (with validation), update, and delete product details.

**Question Paper 10**

1. **(JavaScript, AJAX & PHP)**  
   Build a **real-time character counter** for a text area field. Show the count dynamically using JavaScript.
2. **(Laravel)**  
   Create a **basic event management system** where users can add and delete events with dates and descriptions.

**Question Paper 11**

1. **(JavaScript, AJAX & PHP)**  
   Develop a Registration, login and logout system using Session.

**(Laravel)**  
Develop a **simple expense tracker** where users can:

1. Add(with validation), edit, and delete daily expenses.
2. Categorize expenses (e.g., Food, Travel, Shopping).
3. View a summary of total expenses

**Question Paper 12**

1. **(JavaScript, AJAX & PHP)**  
   Create a **feedback form** where users enter their feedback(with validation), and it is submitted using AJAX without reloading the page.
2. **(Laravel)**  
   Implement a **simple online book library** where users can view a list of books, add(with validation), remove books.

**Question Paper 13**

1. **(JavaScript, AJAX & PHP)**  
   Build a **real-time search bar** where users type a product name, and matching results appear dynamically.
2. **(Laravel)**  
   Develop a **simple job listing platform** where employers (separate login) can post jobs and job seekers (anyone, public page) can view them.

**Question Paper 14**

1. **(JavaScript, AJAX & PHP)**  
   Implement a **basic weather app** that fetches temperature and weather conditions using an API and displays it dynamically.
2. **(Laravel)**  
   Create a **basic movie database** where users can add, edit, and delete movie records.(website creation)

**Question Paper 15**

1. **(JavaScript, AJAX & PHP)**  
   Build a **live comment section** where users can post comments, and new comments appear dynamically using AJAX.
2. **(Laravel)**  
   Develop a **basic invoice management system** where users can add customer details and generate invoices.

**Question Paper 16**

1. **(JavaScript, AJAX & PHP)**  
   Create a **"like" button** for blog posts where clicking it updates the count dynamically. Display the count on the screen.
2. **(Laravel)**  
   Implement a **simple personal finance tracker** where users can log their daily expenses.

**Question Paper 17**

1. **(JavaScript, AJAX & PHP)**  
   Build a **basic task manager** where users can add tasks and mark them as completed.
2. **(Laravel)**  
   Develop a **file upload system** where users can upload a document (only pdf, with validation), stores the file names in DB and display filenames in UI.

**Question Paper 18**

1. **(JavaScript, AJAX & PHP)**  
   Create a **poll system** where users vote on an option, and the total votes update dynamically.
2. **(Laravel)**  
   Develop a **basic student record system** where users can add, edit, and delete student details.

**Question Paper 19**

1. **(JavaScript, AJAX & PHP)**  
   Implement a **real-time chat system** of two pages.
2. **(Laravel)**  
   Build a **basic appointment booking system** where users can book appointments and view them.

**Question Paper 20**

1. **(JavaScript, AJAX & PHP)**  
   Create a **simple e-commerce cart** where users can add items, and the total updates dynamically.
2. **(Laravel)**  
   Develop a **basic feedback system** where users can submit feedback, and admins can view them.

**Question Paper 21**

1. **(JavaScript, AJAX & PHP)**  
   Create a **contact form** where users enter their name, email, and message. The form should submit data using AJAX without reloading the page.
2. **(Laravel)**  
   Develop a **basic student registration system** where admin can add, edit, and delete student details.

**Question Paper 22**

1. **(JavaScript, AJAX & PHP)**  
   Build a **real-time word counter** where users type in a text box, and the word count updates dynamically (use AJAX and PHP).
2. **(Laravel)**  
   Implement a **basic inventory management system** where users can add, update, and delete product details.

**Question Paper 23**

1. **(JavaScript, AJAX & PHP)**  
   Create a **simple to-do list** where users can add and remove tasks dynamically using AJAX.
2. **(Laravel)**  
   Develop a **file uploading system.**

**Question Paper 24**

1. **(JavaScript, AJAX & PHP)**  
   Implement a **real-time typing indicator** where users see a "User is typing..." message when someone types in a box.
2. **(Laravel)**  
   Build a **basic expense tracker** where users can log their expenses and **view a summary**.

**Question Paper 25**

1. **(JavaScript, AJAX & PHP)**  
   Create a **basic rating system** where users rate a product from 1 to 5.
   * 1. **Display 5 Products** → Users select a product.
     2. **Get Username & Feedback** → Users enter their name and give a rating (1-5 stars).
     3. **Store Feedback in Database** → Save the data using PHP & MySQL.
     4. **Retrieve & Display Feedback** → Show feedback dynamically using AJAX.
2. **(Laravel)**  
   Develop a **simple book catalog system** where users can add, update, and delete books.

**Question Paper 26**

1. **(JavaScript, AJAX & PHP)**  
   Create a **basic blog system** where users can write, and delete text posts.
2. **(Laravel)**  
   Build a **basic image gallery** where users can upload images, and view images.

**Question Paper 27**

1. **(JavaScript, AJAX & PHP)**  
   Implement a **simple email subscription form** where users enter their email, and it gets saved in the database via AJAX.
2. **(Laravel)**  
   Implement a **simple email subscription form** where users enter their email, and it gets saved in the database via AJAX. Send a confirmation mail to them.

**Question Paper 28**

1. **(JavaScript, AJAX & PHP)**  
   Create a **live search function** where users search for an item, and results appear dynamically without page reload.
2. **(Laravel)**  
   Implement a **basic customer feedback system** where users submit feedback, and admins can view them.

**Question Paper 29**

1. **(JavaScript, AJAX & PHP)**  
   Build a **simple shopping cart** where users can add products, and the cart updates dynamically using AJAX.
2. **(Laravel)**  
   Develop a **basic attendance system** where teachers mark student attendance, and students can check their records. Student’s list should be displayed. Teacher can mark attendance based on the date.

**Question Paper 30**

1. **(JavaScript, AJAX & PHP)**  
   Implement a **basic currency converter**.
2. **(Laravel)**  
   Create a **simple hotel room booking system** where users can check availability and reserve rooms.