

# **Project Report: Task Manager Application with Laravel, Bootstrap, AJAX, Postman, and Mailtrap**

## **Abstract**

This project introduces a Task Manager Application developed using Laravel, Bootstrap, AJAX, Postman, and Mail trap. The application streamlines task management, offering a user-friendly interface for creating, tracking, and receiving email reminders for tasks.

The primary goal of this project is to showcase the integration of modern web development technologies in building a comprehensive Task Manager Application. This application serves as a practical example of efficient task organization and productivity enhancement.

In an evolving digital landscape, projects like this exemplify the role of web-based solutions in simplifying daily workflows and boosting productivity. Whether for individual task management or collaborative projects, this application demonstrates the power of web technology in our lives.

By embracing Laravel, Bootstrap, AJAX, Postman, and Mail trap, this project provides a valuable learning experience for web developers seeking to create cutting-edge applications that make tasks more manageable and enhance productivity.

## Objective

The primary objectives of this project are as follows:

1. **Development Showcase:** Create a functional Task Manager Application that serves as a compelling demonstration of web development technologies, highlighting the seamless integration of Laravel, Bootstrap, AJAX, Postman, and Mail trap.
2. **Efficient Task Management:** Design an intuitive and user-friendly interface that allows users to efficiently create, update, categorize, and track tasks, thereby enhancing personal and professional task management.
3. **Learning Resource:** Offer a valuable learning resource for web developers, enabling them to explore the application's architecture, codebase, and best practices in modern web development, from front-end design to back-end authentication and email integration.
4. **Real-World Relevance:** Create a practical solution that addresses the real-world need for effective task management tools, catering to both individual users and collaborative teams.
5. **Technological Proficiency:** Encourage developers to become proficient in utilizing Laravel, Bootstrap, AJAX, Postman, and Mail trap, empowering them to apply these technologies effectively in their own web projects.

By achieving these objectives, this project aims to contribute to the understanding and utilization of modern web development tools and practices, ultimately facilitating improved task management and productivity.

# **Introduction**

## **The Significance of Task Management**

Effective task management is a cornerstone of personal and professional productivity. In an era defined by digital transformation and an ever-accelerating pace of work, the ability to organize, track, and prioritize tasks efficiently has become paramount. Task management tools have evolved from simple to-do lists to sophisticated web applications that cater to a diverse range of user needs.

## **The Need for Modern Task Management Solutions**

The demands of modern life and work necessitate task management solutions that offer flexibility, accessibility, and robust functionality. Today's tasks span personal goals, professional projects, and collaborative efforts, requiring tools that can adapt to various contexts. Web-based task management applications have emerged as indispensable solutions, offering the advantages of accessibility from anywhere, real-time collaboration, and integration with other digital tools.

## **The Role of Web Development in Task Management**

Web development plays a pivotal role in shaping the landscape of task management. It empowers developers to create web applications that streamline task organization, enhance user experience, and provide innovative features. These applications bridge the gap between traditional to-do lists and comprehensive task management platforms, catering to a diverse user base.

## **The Task Manager Application Project**

The Task Manager Application project is a response to the evolving needs of task management in the digital age. It is an endeavor to design, develop, and present a task management solution that leverages the capabilities of modern web development technologies. This project showcases the integration of Laravel, Bootstrap, AJAX, Postman, and Mailtrap to create a robust, user-friendly, and feature-rich application.

## **The Evolution of Task Management**

Task management has undergone a significant transformation in recent years. What was once a process managed with pen and paper or basic to-do lists has evolved into a dynamic digital landscape. The demand for efficient task organization and collaboration has given rise to a new generation of task management tools, and web-based applications are at the forefront of this evolution.

## **The Power of Web-Based Solutions**

Web-based task management solutions have revolutionized the way individuals and teams organize their workloads. These applications offer a wide range of features, from creating tasks and setting deadlines to categorizing work and sharing responsibilities. The accessibility of web-based tools allows users to manage tasks from anywhere, promoting flexibility and productivity.

## Methodology

### Technologies Used

- **Laravel:** Laravel is a robust PHP web application framework known for its elegant syntax and powerful features. It simplifies complex tasks such as routing, authentication, and database interaction.
- **Bootstrap:** Bootstrap is a popular front-end UI library that streamlines web design and development. It provides responsive components and a clean, modern design.
- **AJAX:** Asynchronous JavaScript and XML (AJAX) is used for making asynchronous API calls to the server, enabling dynamic updates without page refresh.
- **Postman:** Postman is a popular API development and testing tool that simplifies the process of designing, testing, and documenting APIs by providing an intuitive user interface for sending HTTP requests, viewing responses, and automating various aspects of API development and testing.
- **Mail trap:** Mail trap is a service for testing email delivery in a development environment. It allows developers to capture and inspect emails sent from their applications.

### Application Features

The Task Manager Application boasts the following features:

- Task Creation, Update, and Deletion.
- Task Status Tracking (e.g., In Progress, Completed).
- Email Reminders for Upcoming Tasks using Mailtrap.

## Code

### Laravel Code Samples

```
public function create(Request $request)
{
    // dd($request->all());
    $data = new TaskManager();
    $data->task_description = $request->get('task_description');
    $data->task_owner = $request->get('task_owner');
    $data->task_owner_email = $request->get('task_owner_email');
    $data->task_eta = $request->get('task_eta');

    if ($data->save()) {
        dispatch(new SendEmailJob($data));
        return "Task saved successfully";
    } else {
        return "Failed to save Task";
    }
}

public function index()
{
    $data = TaskManager::get();
    return $data;
}

public function getTaskById($id)
{
    $data = TaskManager::find($id);
    return $data;
}

public function update(Request $request, $id)
{
    $data = TaskManager::find($id);
    $data->task_description = $request->get('task_description');
    $data->task_owner = $request->get('task_owner');
    $data->task_owner_email = $request->get('task_owner_email');
    $data->task_eta = $request->get('task_eta');
    if ($data->save()) {
        return "Task updated successfully";
    } else {
        return "Failed to update Task";
    }
}

public function markAsDone($id)
{
    $data = TaskManager::find($id);
    $data->status = 1;

    if ($data->save()) {
        dispatch(new SendEmailJob($data));
        return "Task Marked as Done";
    } else {
        return "Failed to Mark Task as done";
    }
}
```

```

    }
}
public function delete($id)
{
    public function create(Request $request)
    {
        // dd($request->all());
        $data = new TaskManager();
        $data->task_description = $request->get('task_description');
        $data->task_owner = $request->get('task_owner');
        $data->task_owner_email = $request->get('task_owner_email');
        $data->task_eta = $request->get('task_eta');

        if ($data->save()) {
            dispatch(new SendEmailJob($data));
            return "Task saved successfully";
        } else {
            return "Failed to save Task";
        }
    }
}
public function index()
{
    $data = TaskManager::get();
    return $data;
}
public function getTaskById($id)
{
    $data = TaskManager::find($id);
    return $data;
}
public function update(Request $request, $id)
{
    $data = TaskManager::find($id);
    $data->task_description = $request->get('task_description');
    $data->task_owner = $request->get('task_owner');
    $data->task_owner_email = $request->get('task_owner_email');
    $data->task_eta = $request->get('task_eta');
    if ($data->save()) {
        return "Task updated successfully";
    } else {
        return "Failed to update Task";
    }
}
}
public function markAsDone($id)
{
    $data = TaskManager::find($id);
    $data->status = 1;

    if ($data->save()) {
        dispatch(new SendEmailJob($data));
        return "Task Marked as Done";
    } else {
        return "Failed to Mark Task as done";
    }
}
}

```

```

}
public function delete($id)
{
    $data = TaskManager::find($id);

    if ($data->delete()) {
        return "Task deleted successfully";
    } else {
        return "Failed to delete Task";
    }
}

$data = TaskManager::find($id);

if ($data->delete()) {
    return "Task deleted successfully";
} else {
    return "Failed to delete Task";
}
}

```

### Creation of Task sample

```

<div class="modal fade" id="createTaskModal" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Add Task</h5>
        <button type="button" class="btn-close" data-bs-
dismiss="modal"></button>
      </div>
      <div class="modal-body">
        <form action="">
          <div class="form-group">
            <label for="createTaskDescription">Task Description</label>
            <input type="text" class="form-control"
id="createTaskDescription" placeholder="Enter Task Description">
          </div>
          <div class="form-group">
            <label for="createTaskOwner">Task Owner</label>
            <input type="text" class="form-control" id="createTaskOwner"
placeholder="Enter Task Owner">
          </div>
          <div class="form-group">
            <label for="createTaskEmail">Task Owner Email</label>
            <input type="text" class="form-control" id="createTaskEmail"
placeholder="Enter Task Owner Email ">
          </div>
          <div class="form-group">
            <label for="createTaskETA">Task ETA</label>
            <input type="date" class="form-control" id="createTaskETA"
placeholder="Enter Task ETA">
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```



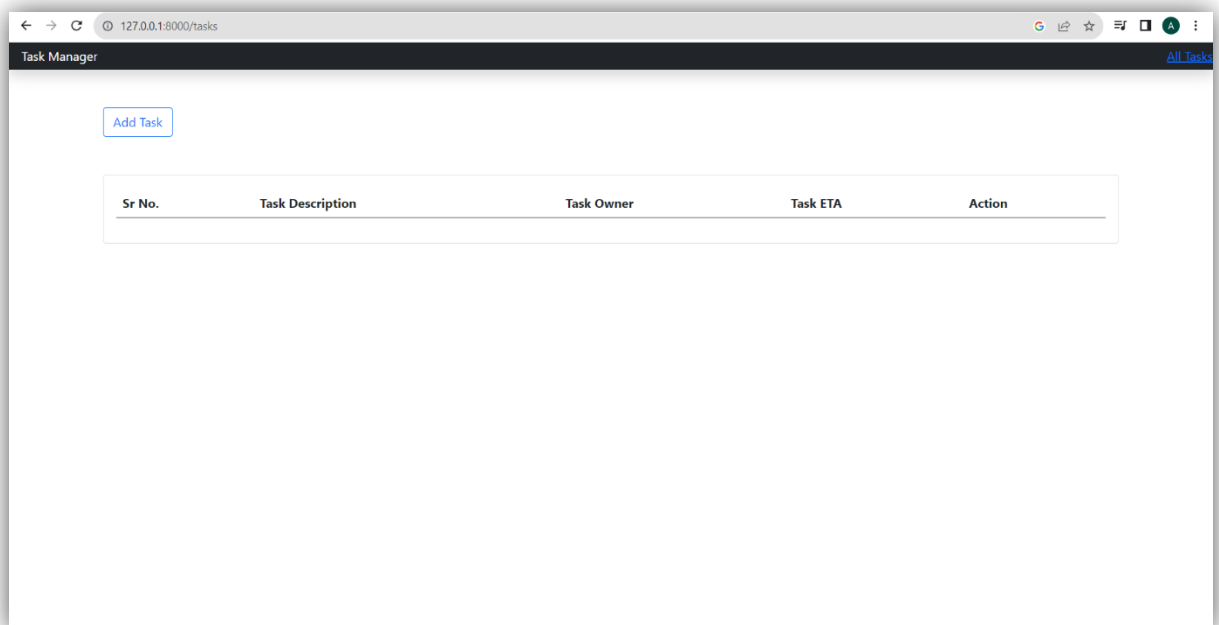
```

        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Close</button>
            <button type="button" class="btn btn-primary"
onclick="createTask()">Submit</button>
        </div>
    </div>
</div>
</div>

```

## Screenshots of Results and Output

Figure 1: Task List Interface



**Figure 2: Task Creation**

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/tasks'. The page title is 'Task Manager'. A modal window titled 'Add Task' is open in the center. The modal contains the following fields:

- Task Description:** A text input field containing 'Task 1'.
- Task Owner:** A text input field containing 'owner1'.
- Task Owner Email:** A text input field containing 'owner1@yomail.com'.
- Task ETA:** A date input field containing '16-05-2023'.

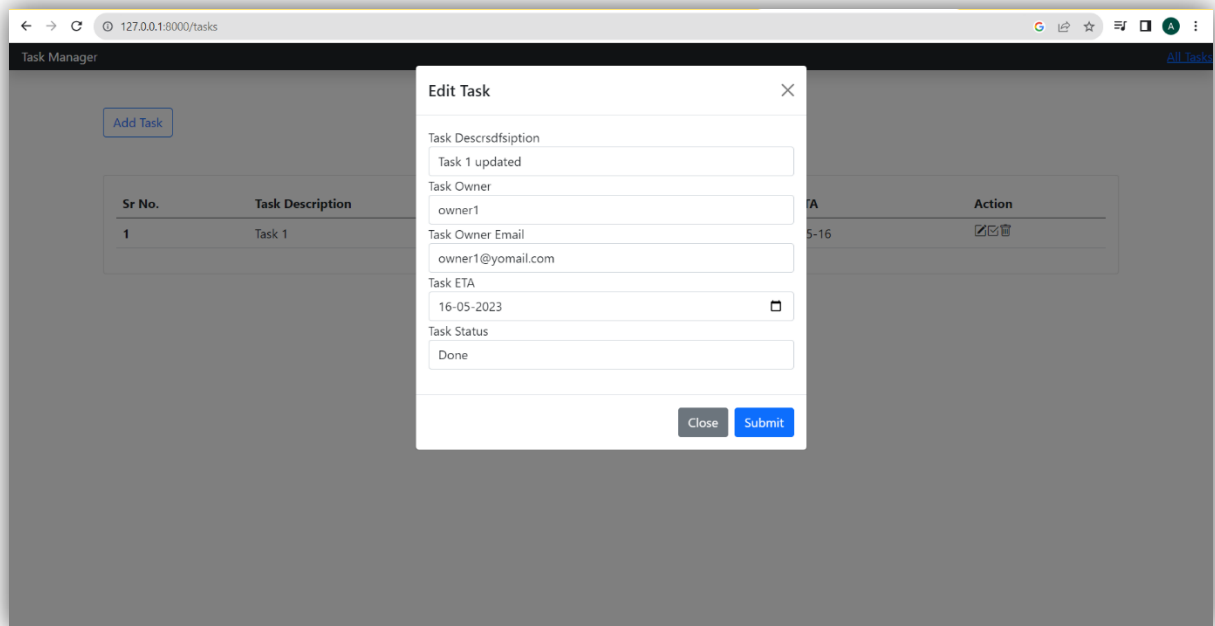
At the bottom of the modal, there are two buttons: 'Close' and 'Submit'. In the background, a table with the following headers is visible: 'Sr No.', 'Task Description', 'Task ETA', and 'Action'. An 'Add Task' button is also visible in the top left of the page.

The screenshot shows the same web browser window, but the 'Add Task' modal is closed. The 'Task Manager' page now displays a table with the following data:

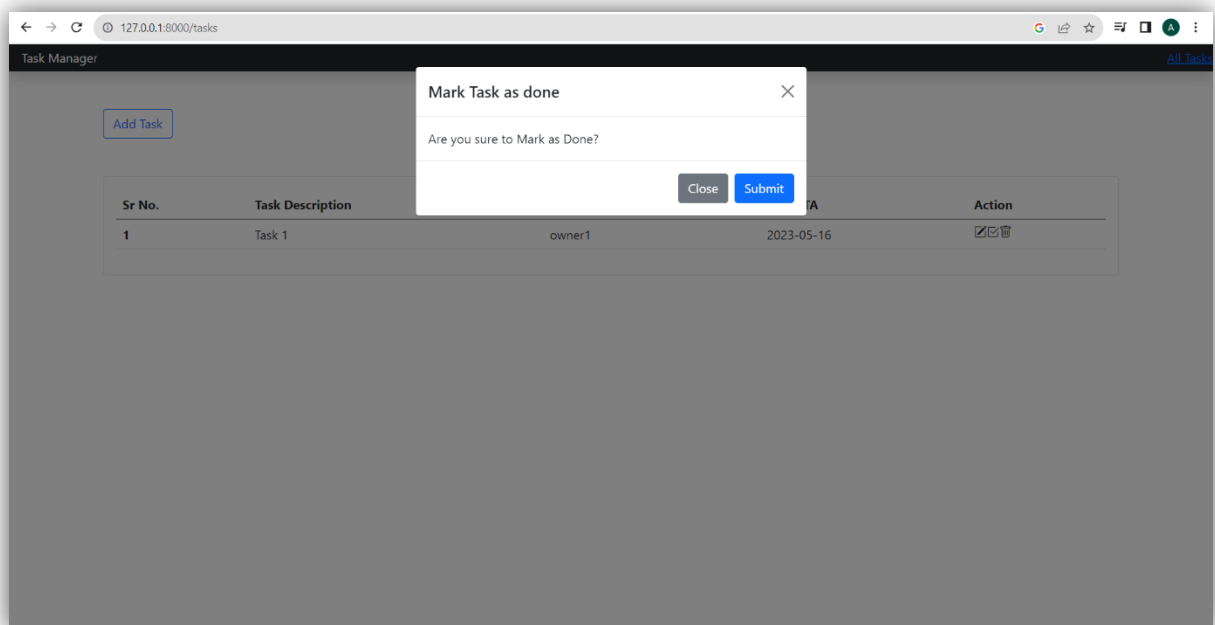
Sr No.	Task Description	Task Owner	Task ETA	Action
1	Task 1	owner1	2023-05-16	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

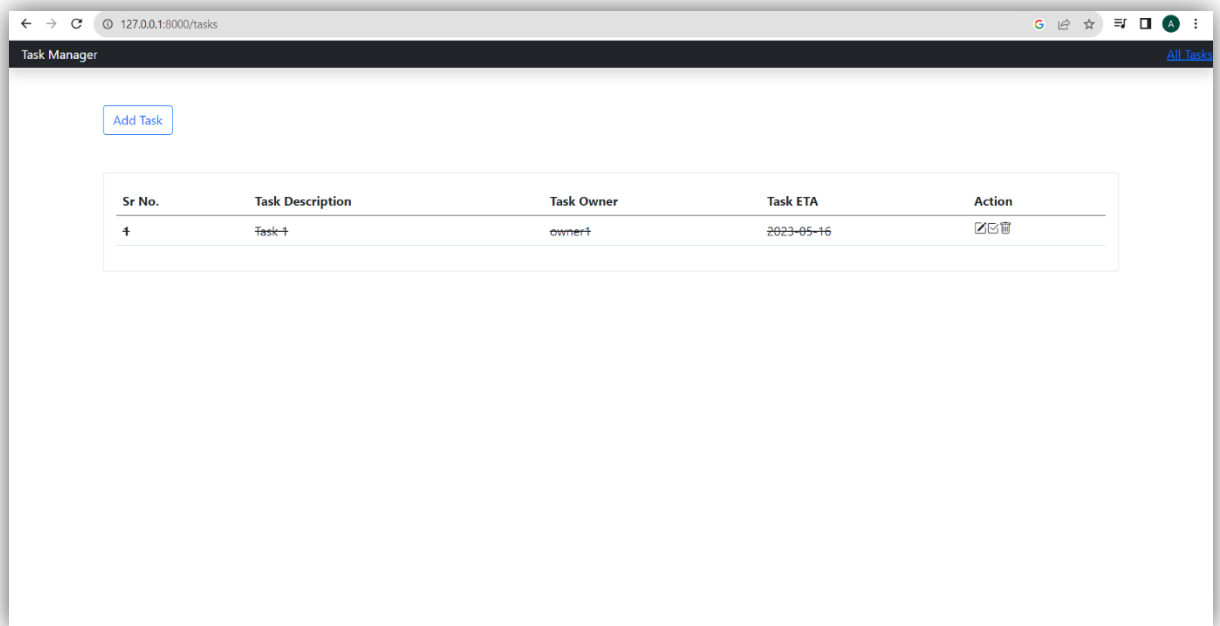
The 'Action' column contains three icons: a checked checkbox, an unchecked checkbox, and a trash icon. An 'Add Task' button is located in the top left corner of the page.

**Figure 3: Update Task**

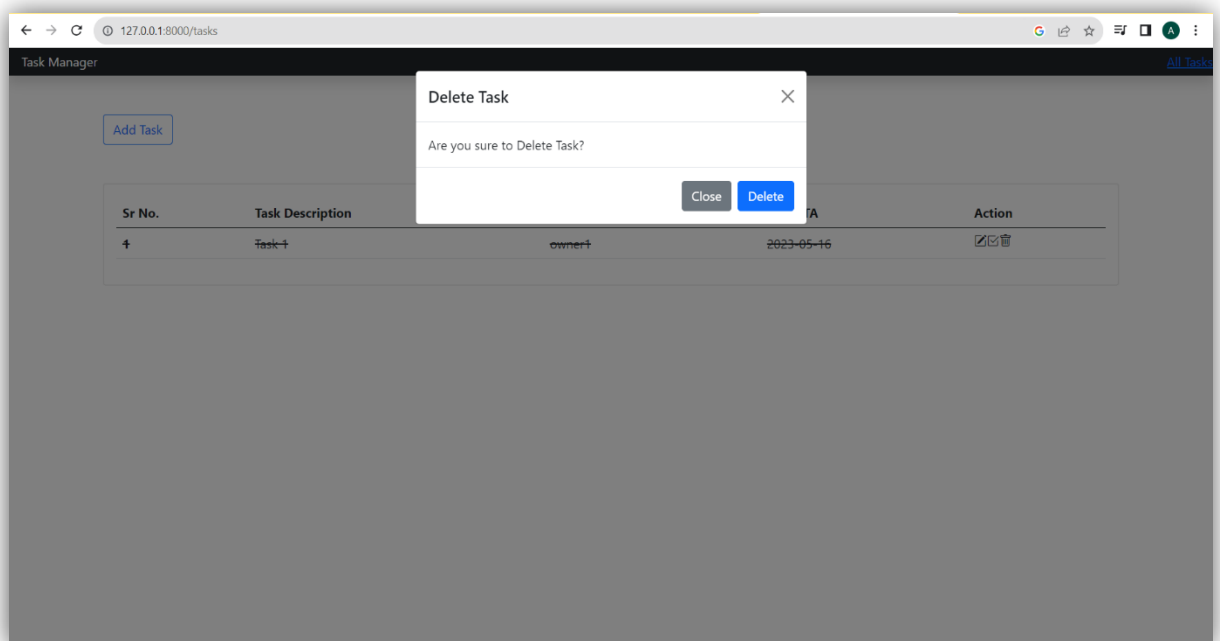


**Figure 4: Mark as Done Task**





### Figure 5: Delete Task



## **Conclusion**

In conclusion, the Task Manager Application demonstrates the power and flexibility of Laravel, Bootstrap, AJAX, Postman, and Mail trap in creating a modern, feature-rich web application. Users can efficiently manage tasks, categorize them, track their status, and receive email reminders for impending deadlines.

This project provides valuable insights into building a secure, responsive, and functional web application, suitable for both personal and professional task management. Additionally, it underscores the importance of staying organized and leveraging technology to enhance productivity.

The combination of Laravel's robust back-end capabilities and Bootstrap's sleek front-end design makes this application a versatile tool for task management needs. The integration of AJAX ensures a smooth user experience, while Mail trap contribute to the security and reliability of the application.

In the ever-evolving landscape of web development, projects like this Task Manager Application pave the way for innovative solutions that simplify our daily lives and empower us to work more efficiently.