KillTasks

Task Management System Design Document

Table of Contents

[Introduction 2](#_Toc163290350)

[Goal 2](#_Toc163290351)

[Dashboard 2](#_Toc163290352)

[Personal 3](#_Toc163290353)

[Settings 3](#_Toc163290354)

[Non-Goal 3](#_Toc163290355)

[Design Detail 3](#_Toc163290356)

[Architecture 3](#_Toc163290357)

[Database Design 4](#_Toc163290358)

[Conceptual 4](#_Toc163290359)

[Physical 4](#_Toc163290360)

[Frontend Technology Stack 4](#_Toc163290361)

[Responsible 4](#_Toc163290362)

[Summary 5](#_Toc163290363)

# 

# Introduction

In modern society, we are often expected to be multi-tasking but in reality, just like a computer or a CPU, we are only context-switching between each other, to keep ourselves on track.

Task management is a handy tool for personal use or groups to keep track of the status of different tasks, the status, and who handles each task.

# To Start

1. First, import the “killtasks” folder into the “htdocs” in the “Xampp”
2. Go to http://localhost/phpMyAdmin then create database “killtasks”; copy, paste, and run the DDL.sql from the “database” directory, after successfully creating tables and inserting dummy data
3. Go to <http://localhost/killtasks>
4. Login with email: [admin@admin.com](mailto:admin@admin.com) password: [admin@admin.com](mailto:admin@admin.com).
5. Try out other functions.

# Goal

Our task management is to have a clear view of all the tasks, here is the list of functionalities the task management should be able to perform.

1. Show all tasks
2. Categorizing by status or group
3. A dashboard to oversee status and tasks.
4. The task list can be filtered by team.
5. Description, title, or responsible personnel can search for tasks.
6. User accounts can be created and updated.
7. Support user authentication
8. Creating, Retrieve, Update, Delete tasks.

The main goal of our task management is to give the user a quick, clean and clear view of the breakdowns of all the current tasks, so he/she will be on track with all ongoing responsibilities, and hopefully get ahead of the rest.

We believe the first stage of this application will have 4 main pages, a login page, the dashboard, a personal task page, and an account settings page.

## Dashboard

The dashboard with a table to show all tasks, statuses, and categories, we will also include a chart of who has the most responsibilities, as well as the ratio of tasks that have been completed, working, and delayed, the user can also update his/her task, and upcoming incomplete tasks.

## Personal

The personal task page will allow the user to view the details of his/her task, and able to view the history of completed tasks.

## Settings

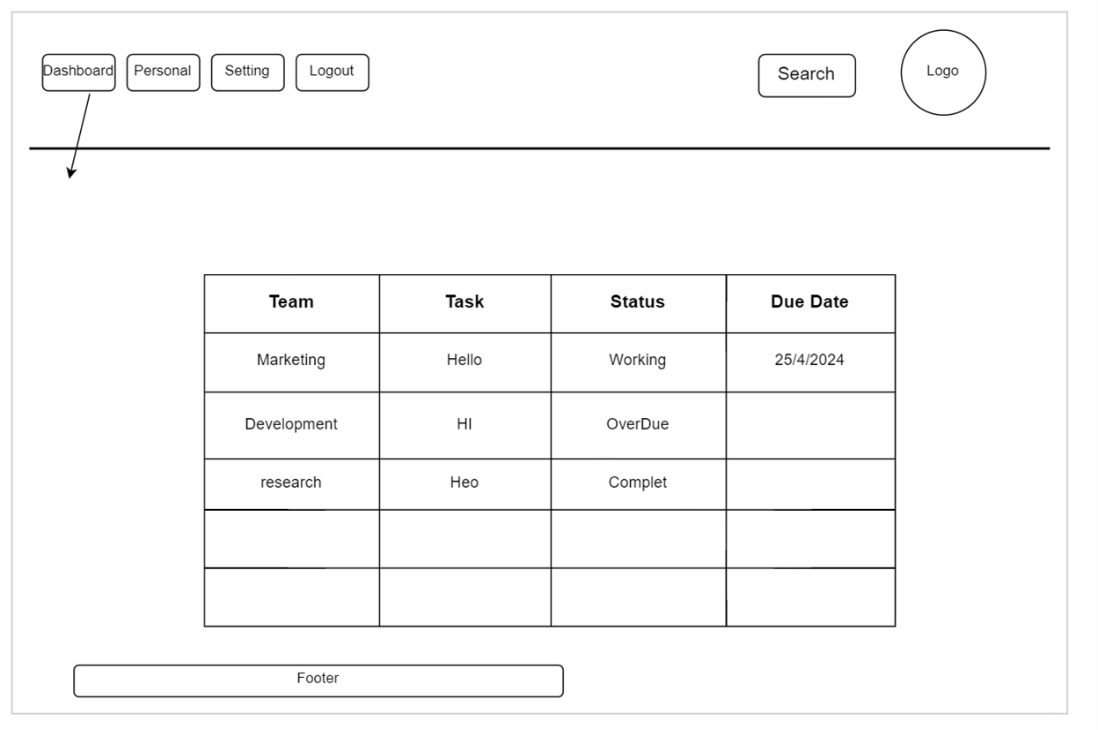
For updating the account information except email.

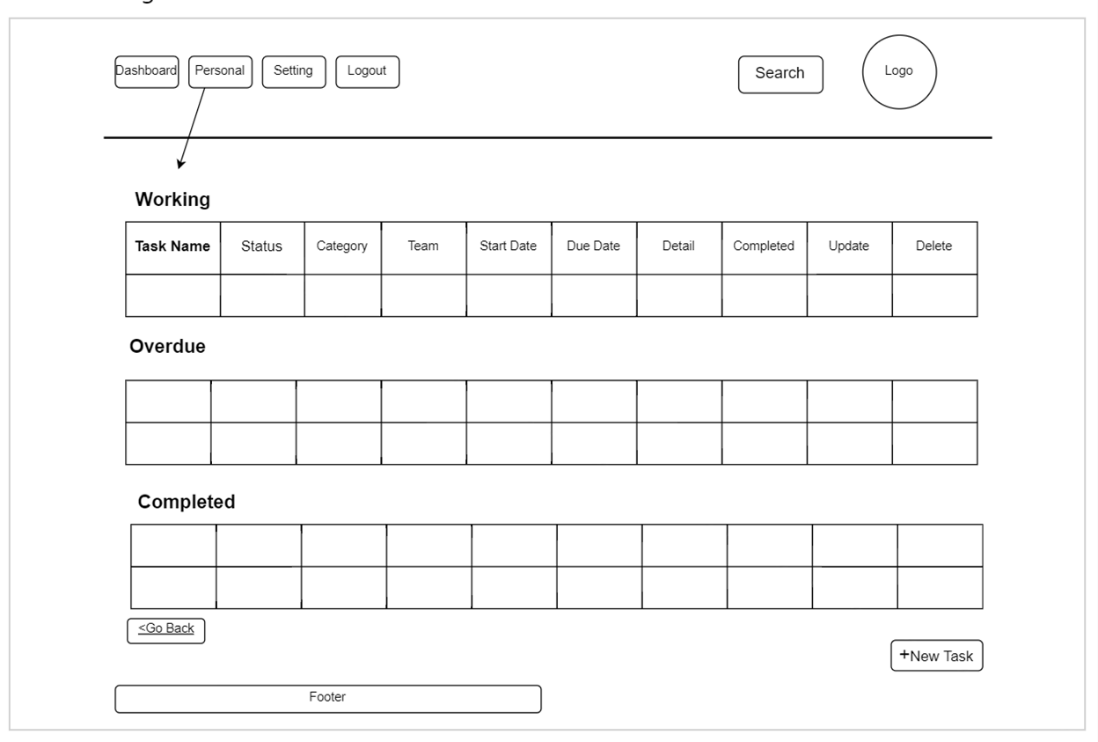
# Non-Goal

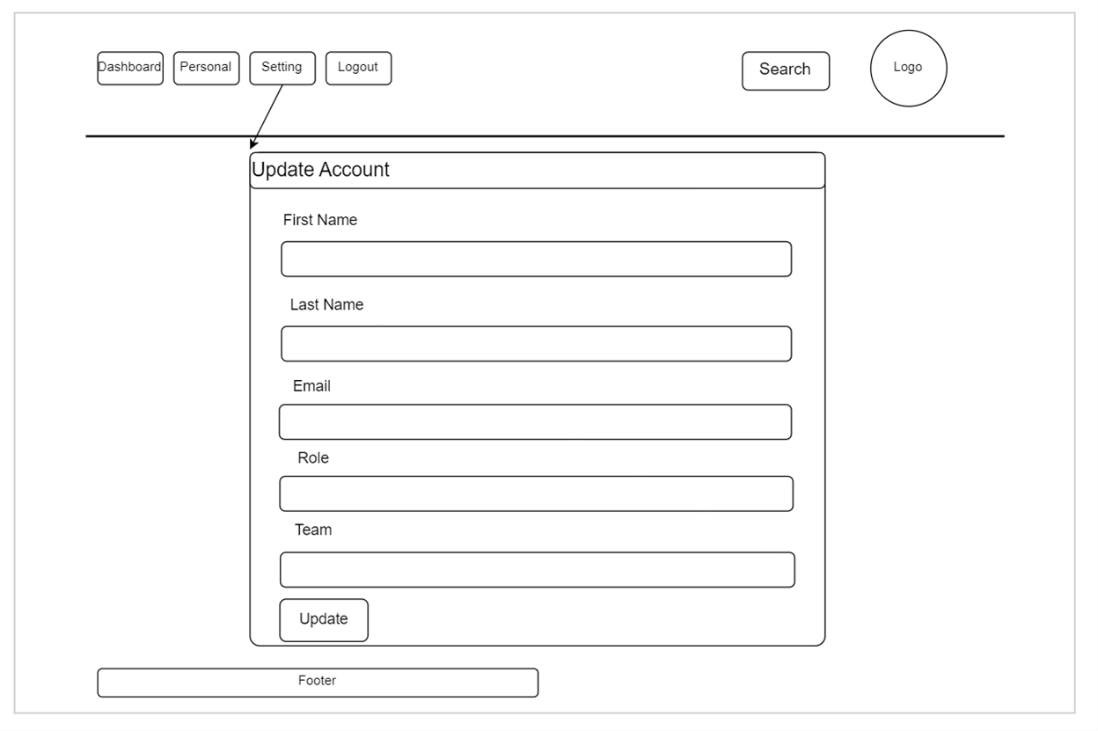
The task management system is not meant to include a calendar, Gantt chart, or Kanban board, it also does not support email inbox, real-time messaging, and not supporting hierarchical roles.

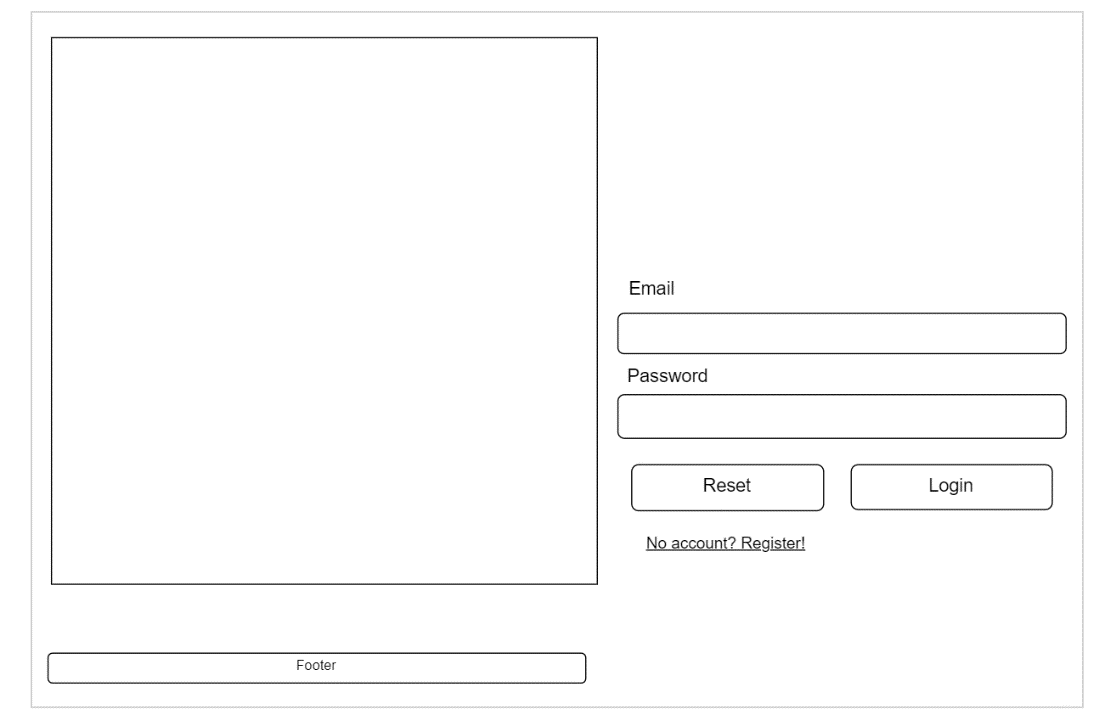
# Design Detail

## Wireframe

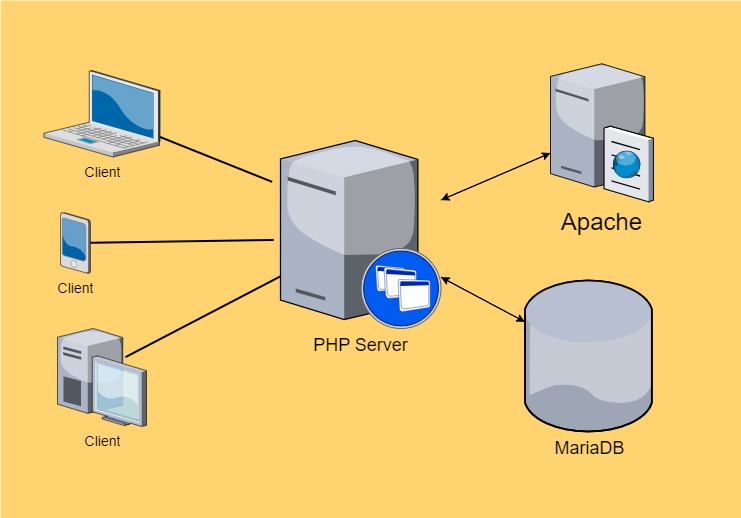








## Architecture

The application will be using PHP, MariaDB, and Apache as the server-side technology stack. The pages will be dynamically loaded and updated with PHP, JavaScript, HTML5 and CSS3. The application will be server-side rendered to provide a monolithic application approach.

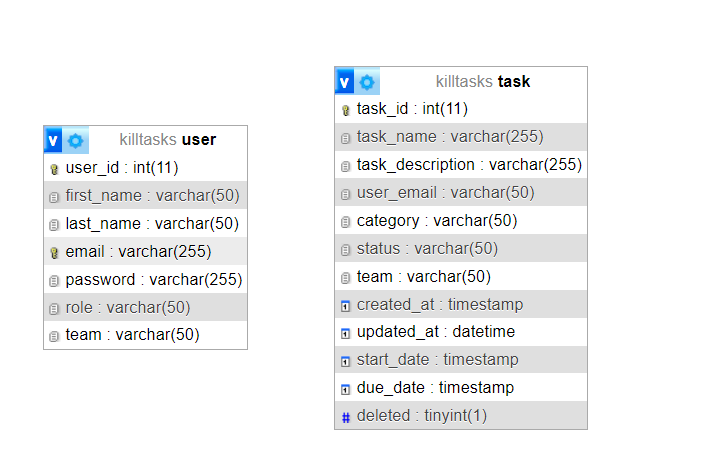
## Database Design

### Conceptual

User – first\_name, last\_name, email, password, role, team, created\_at

Task – task\_name, task\_description, status, user, team, created\_at, updated\_at, start\_date, due\_date

### Physical



### Frontend Technology Stack

For KillTasks, the frontend technology stack will be HTML5, CSS3, JavaScript (ES2022), Bootstrap.

### Directory Definition

* **Auth, Private, and Public** are the client-facing pages.
  + **JS** includes all JavaScript
  + **CSS** includes all stylings
  + **Partials** holds the components for reusable.
* **Server** are all the server side code
  + **Service** includes all business logics
  + **Config** includes PDO connection with MySQL(MariaDB) Database
  + **Model** includes the data objects for User, Task and Response in class
  + **Utils** includes help functions for converting, sanitizing, etc.

## Responsible

David

* Database design, Backend, Mock data
* All server code
* Dashboard

Durra

* Wireframe
* Account Setting

Candice

* Nav
* Footer
* Header
* Personal

For the team tasks checklist please find the checklist.xlsx file.

# Summary

The task management application is a lightweight task management for personal and small groups to use, the user can CRUD tasks, and be able to have an overview of the overall status of all tasks, and performance.