Ministry of Higher Education and Scientific Research
Hassiba Benbouali University of Chlef
Faculty of Exact Sciences and Computer Science
Computer Science department

Report

SOFTWARE ENGINEERING AND PROJECT ENGINEERING

Specialty: Computer Science

Team Leader
TAHRI YOUCEF

Theme:

Tasks Management Software

The members of this project:

MAHALEEL ALI SAFI RAHMAN	BOUMARAF MOUAD ABDERRAHMANE	TIATOUCHINE HAMZA
AMINE SELLAI	MESSAFA.MOHAMMED	FETHI MEZIANE
ABDELALI SID AHMED	ZAKARYA MEDDAHI	TAHRI YOUCEF

Academic Year 2024-2025

Abstract

This report discusses the importance of task management software in the modern work environment. The paper describes some of the main features that include creating, prioritizing, sharing, and tracking of tasks. Various advantages are discussed on the use of task management software: it improves productivity, enhances collaboration, offers better project management, and gives more organization. It also underlines some popular options available on the market and guides on how to choose suitable software that will satisfy particular needs. Properly used task management software enables individuals and teams to be more effective, productive, and more successful in everything they try.

Introduction

Task management is central to the success of any organization, team, or individual. Tasks are the very building blocks of projects and organizational workflows; without an efficient system to manage them, inefficiencies and miscommunications are likely to arise. The "Tasks Management Software" project comes to address the increasing need for an efficient, user-friendly, and robust system that streamlines task allocation, tracking, and collaboration. This report provides a detailed overview of the project, its objectives, features, system architecture, development methodology, and the benefits expected.

The rising complexity of today's work environments, along with teams geographically dispersed, has made traditional task management tools inadequate. This software is envisioned as a holistic solution, combining simplicity, powerful features, and scalability to cater to diverse user needs, whether for small teams, large enterprises, or individual freelancers. By facilitating effective task planning, monitoring, and collaboration, the project is aimed at unlocking higher levels of productivity and efficiency.

Objectives

The major objectives of the Tasks Management Software are to:

Improve Productivity: It gives a structured way in managing the tasks in such a manner that the user can focus on their time and work on high-value activities.

Improve Collaboration: Facilitate improved communication and collaboration among team members with integrated tools and shared platforms.

Real-Time Tracking: To offer live updates, progress monitoring, and updating of stakeholders on the status of different tasks and projects.

Actionable Insights: It provides analytics and reporting for making better decisions and understanding where improvements need to be made.

Cross-Platform Accessibility: Ensuring ease of use and accessibility on desktops, mobile devices, and tablets to help users in varied environments.

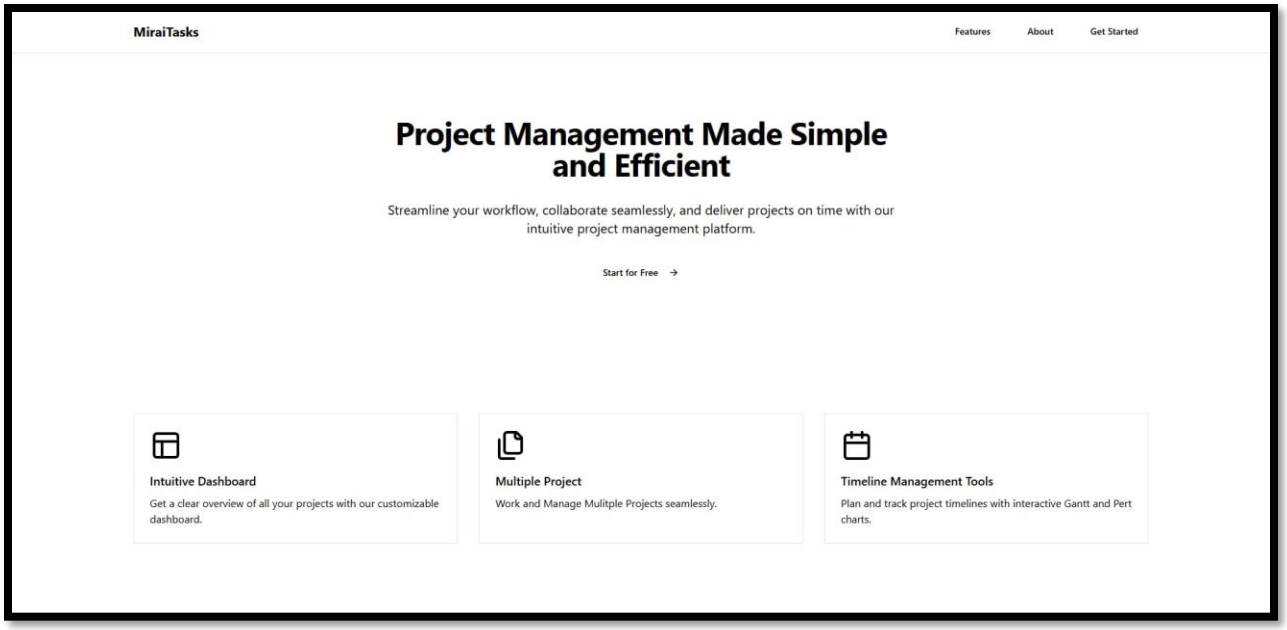
These objectives represent the major mission of the software in equipping the user with the means of manipulating their workload in the most effective way, whatever its complexity or the size of the team. Reaching such goals means striking a balance between simplicity and rich functionality while still allowing customizations.

Features and Functionalities

Core Features:

1. **Task Creation and Assignment:** Users can create tasks with detailed descriptions, set priorities and deadlines, and assign them to specific team members or groups. Tasks can also include dependencies to indicate prerequisite activities that must be completed first.
2. **Progress Tracking:** Tools such as Kanban boards, Gantt charts, and customizable progress indicators allow users to visualize workflows and track the status of each task in real-time.
3. **Notifications and Alerts:** Automated reminders help users stay on top of deadlines and receive updates about changes or new assignments. Notifications are customizable to align with individual preferences.
4. **Collaboration Tools:** Teams can collaborate seamlessly using built-in commenting systems, file sharing, and instant messaging. The software also logs all interactions to maintain a clear communication history.
5. **User Roles and Permissions:** Role-based access ensures that team members only see information and features relevant to their roles, thereby enhancing security and usability.

Front-End Design and User Interface Overview of the MiraiTasks Application

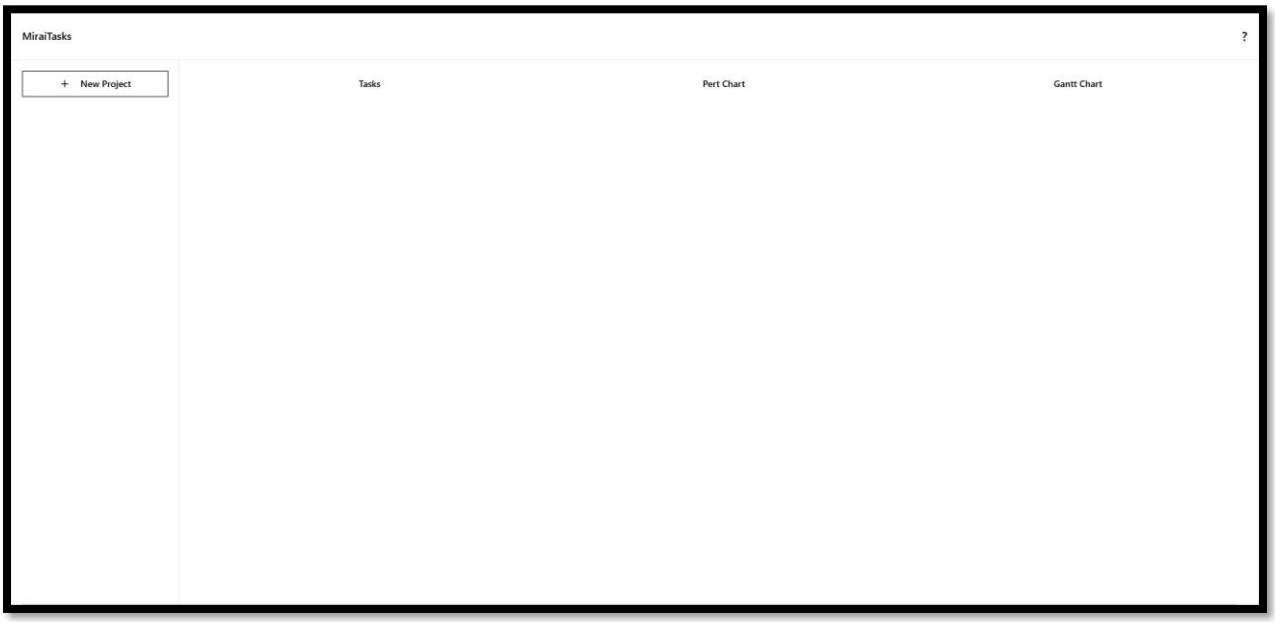


Features:

Intuitive Dashboard: The application includes an intuitive dashboard that shows the overview of all projects. This feature is very important for users who have to manage a lot of tasks and projects at the same time. The dashboard is customizable, meaning it can be tailored to meet specific needs.

Multi-Project Management: The platform has the possibility to manage multiple projects with no breaks. This will help a group or individual that is working on several tasks relating to different projects; this will ensure that nothing is left out.

Timeline Management Tools: This application contains interactive Gantt and PERT charts for planning and tracking project timelines.
These are helpful in visualizing project schedules and finding critical paths of ensuring timely deliveries of tasks.



Visual Design and Layout:

Clean, modern look and feel: the general design of the page is clean and modern, putting much emphasis on usability. Whitespace, clear typography, intuitive icons all contribute to creating a superior user experience.
Responsiveness: layout is meant to be responsive; it means that an application should be fully functional and accessible o n all ranges of devices from desktops and tablets to smartphones.

User Experience Considerations:

Usability: The design made it more intuitive, reducing the learning curve of the application for inexperienced users. Its usability and clear calls-to-

action let users understand how things should be done on this platform.

Accessibility: In design, accessibility standards are considered so different able ind ividuals could make use of this application. This included considerations for color contrast, font sizes, and keyboard navigation.

This interface is straightforward and user-friendly, facilitating quick and secure access to the application.

New Project: This button allows users to create a new project, facilitating quick initiation of tasks and planning.

Talks: Likely a section for communication or updates related to projects, enabling team collaboration and discussion.

Port Chart: Possibly a typo or shorthand for a PERT chart, which is used for task scheduling and project timeline visualization.

Gantt Chart: a tool for planning and tracking project schedules.

Back-End Architecture and Functionality of the MiraiTasks Application

```jsx
import { useState } from 'react';
import {
  Card,
  CardContent,
  CardDescription,
  CardHeader,
  CardTitle,
} from "@/components/ui/card";
import { Input } from "@/components/ui/input";
import { Button } from "@/components/ui/button";
import { Tabs, TabsContent, TabsList, TabsTrigger } from "@/components/ui/tabs";

const AuthPage = () => {
  const [loginEmail, setLoginEmail] = useState('');
  const [loginPassword, setLoginPassword] = useState('');
  const [signupEmail, setSignupEmail] = useState('');
  const [signupPassword, setSignupPassword] = useState('');
  const [signupConfirmPassword, setSignupConfirmPassword] = useState('');

  const handleLogin = (e) => {
    e.preventDefault();
    // Add login logic here
    console.log('Login:', { email: loginEmail, password: loginPassword });
  };

  const handleSignup = (e) => {
    e.preventDefault();
    // Add signup logic here
    console.log('Signup:', {
      email: signupEmail,
      password: signupPassword,
      confirmPassword: signupConfirmPassword
    });
  };

```

This React component (AuthPage) handles user authentication with login and sign-up forms.

Imports: Uses useState for state management. Imports UI components (Card, Input, Button, Tabs) for building the interface.

State: Manages email, password, and confirm password inputs for both login and sign-up.

Event Handlers:

handleLogin: Logs login credentials (to be replaced with backend logic).

handleSignup: Logs sign-up credentials (to be replaced with backend logic).

```jsx
import React, { useEffect, useState } from "react";
import { Chart } from "react-google-charts";
import { UseData } from "../providers/GraphProvider";

export const options = {
  gantt: {
    criticalPathEnabled: true,
    criticalPathStyle: {
      stroke: "#e64a19",
      strokeWidth: 5,
    },
  },
};

function daysToMilliseconds(days) {
  return days * 24 * 60 * 60 * 1000;
}

export const GanttChart = () => {
  const {projectData} = UseData();
  const [rows, setRows] = useState([]);

  useEffect(() => {
    console.log(rows);
    if (projectData && projectData.nodes && projectData.links) {
      console.log("useEffect Gantt Chart");
      const nodes = projectData.nodes;
      const links = projectData.links;

      const newRows = nodes.map((node, index) => {
        const linkSource = links[index - 1]?.source || null;
        return [
          node.id,
          node.label,
          null,
          null,
          null,
          daysToMilliseconds(node.duration),
          null,
          linkSource,
        ];
      });
      setRows(newRows);
    } else {
      console.warn("Gantt data is undefined or missing required properties");
    }
  }, [projectData]);

  const columns = [
    { type: "string", label: "Task ID" },
    { type: "string", label: "Task Name" },
    { type: "string", label: "Resource" },
    { type: "date", label: "Start Date" },
    { type: "date", label: "End Date" },
    { type: "number", label: "Duration" },
    { type: "number", label: "Percent Complete" },
    { type: "string", label: "Dependencies" },
  ];

```

UI Structure: Uses Tabs to switch between login and sign-up forms.

Login form: Email, password, and a "Login" button.

Sign-up form: Email, password, confirm password, and a "Sign Up" button.

-The second image represents a React component (GanttChart) generates a Gantt chart using the react-google-charts library.

Imports: Uses useEffect and useState for state and side effects. Imports Chart from react-google-charts and UseData from a context provider for data.

Options: Configures the Gantt chart with critical path styling.

Helper Function: daysToMilliseconds: Converts days to milliseconds for chart data.

State and Data Processing:

Fetches project data (nodes and links) from UseData.

Maps nodes to chart rows, including task ID, name, duration, and dependencies.

Columns: Defines the structure of the Gantt chart columns (e.g., Task ID, Task Name, Duration).

Rendering: Uses the Chart component to display the Gantt chart with the processed data.

Conclusion

In conclusion, the development of the MiraiTasks application demonstrates a robust and user-friendly approach to project management, particularly for task scheduling using the PERT method. The front-end design emphasizes simplicity and efficiency, with intuitive navigation, interactive Gantt and PERT charts, and seamless multi-project management. The back-end architecture ensures reliable data processing and integration, supporting dynamic task scheduling and real-time updates. By combining a clean, responsive interface with powerful scheduling tools, MiraiTasks provides a comprehensive solution for teams and individuals to streamline workflows, enhance collaboration, and deliver projects on time. Future enhancements could include advanced analytics, AI-driven task prioritization, and expanded integration capabilities to further elevate the user experience.