Customer Success Platform

Introduction

In the ever-evolving business environment of today, efficient communication and openness are crucial for guaranteeing the triumph of any undertaking. The Customer Success Platform initiative arises as a solution to this requirement, with the goal of transforming how stakeholders are kept informed and involved throughout the project's lifecycle.

Project Overview

The primary goal of this project is to build a resilient Customer Success Platform that automates the dissemination of updates and modifications within our system to pertinent stakeholders. Leveraging advanced technology, we aim to optimize communication pathways, guaranteeing swift and effective delivery of notifications. Ultimately, the platform aims to improve cooperation, transparency, and stakeholder contentment by providing real-time updates on crucial developments.

Key Components of Customer Success Platform:

- **Project Description:** An overview of the project's purpose, goals, and objectives sets the stage for understanding its significance and impact.

- **Scope:** Defining the project's boundaries ensures a clear understanding of what falls within the purview of our efforts.

- **Project Stack (Tech):** This section outlines the technological framework of our project, highlighting the utilization of React.js, PostgreSQL, and Java(Spring-Boot) to accomplish our objectives.

- **Escalation Matrix:** A structured hierarchy and process for escalating issues within the project team ensure timely resolution of concerns.

- **Stakeholders:** Identification of key individuals or groups with a vested interest in the project lays the foundation for effective collaboration and communication.

- **Risk Profiling:** Assessing potential risks and devising strategies for mitigation safeguards our project against unforeseen challenges.

- **Phases/Milestones:** Breaking down the project into manageable phases with specific timelines ensures systematic progress and milestone achievement.

- **Sprint-wise Detail:** Detailed insights into each sprint, including timelines, status, and comments, facilitate efficient project tracking and management.

- **Approved Team:** Listing project team members, their roles, and availability ensures clarity regarding responsibilities and resource allocation.

- **Resources:** Identification of requisite resources, including human capital, equipment, and materials, ensures smooth project execution.

- **Client Feedback:** Documenting client feedback, both positive and negative, facilitates continuous improvement and client satisfaction.

- **Minutes of Meetings:** Recording key discussion points from client meetings ensures alignment and accountability across all stakeholders.

The technological tools employed in our project:

- ## React.js:

  React.js stands at the forefront of our front-end development stack. Renowned for its component-based architecture and virtual DOM rendering, React.js empowers us to build dynamic and interactive user interfaces with unparalleled ease and efficiency. Its declarative syntax and efficient state management mechanisms streamline development workflows, fostering code reusability and maintainability. By leveraging React.js, we aim to deliver a seamless and responsive user experience across a multitude of devices and browsers.

- ## Java (Spring-Boot):

  Spring Boot serves as the cornerstone of our backend infrastructure, providing robust support for server-side Java execution with outstanding performance and scalability. Utilizing a highly efficient, event-driven architecture with non-blocking I/O, Spring Boot optimizes resource utilization and responsiveness, making it an excellent fit for real-time applications similar to ours. Its comprehensive ecosystem of Spring Framework modules and libraries offers a vast array of functionalities and tools, enabling agile development and deployment processes. With Spring Boot, we can effortlessly manage concurrent connections, execute intricate business logic, and seamlessly interact with databases, thereby facilitating resilient and feature-rich server-side operations.

- ## PostgreSQL:

  PostgreSQL stands as our preferred database solution, providing a versatile and scalable platform for data storage and management. As a powerful relational database management system (RDBMS), PostgreSQL offers robust features and capabilities, allowing us to effectively handle structured data. Unlike NoSQL databases, PostgreSQL follows a strict schema-based approach, ensuring data integrity and consistency across tables and relationships. Its ACID-compliant transactions guarantee reliability and durability.

Features

- ## Login using Auth0 And Role Management using Database:

  The Login and Role Management feature, powered by Auth0, serves as the gateway to our Customer Success Platform, providing users with secure access and role-based permissions tailored to their organizational roles. This feature streamlines user authentication and authorization processes, ensuring a seamless and secure user experience. Below, we delve into the details of this feature and its functionalities:

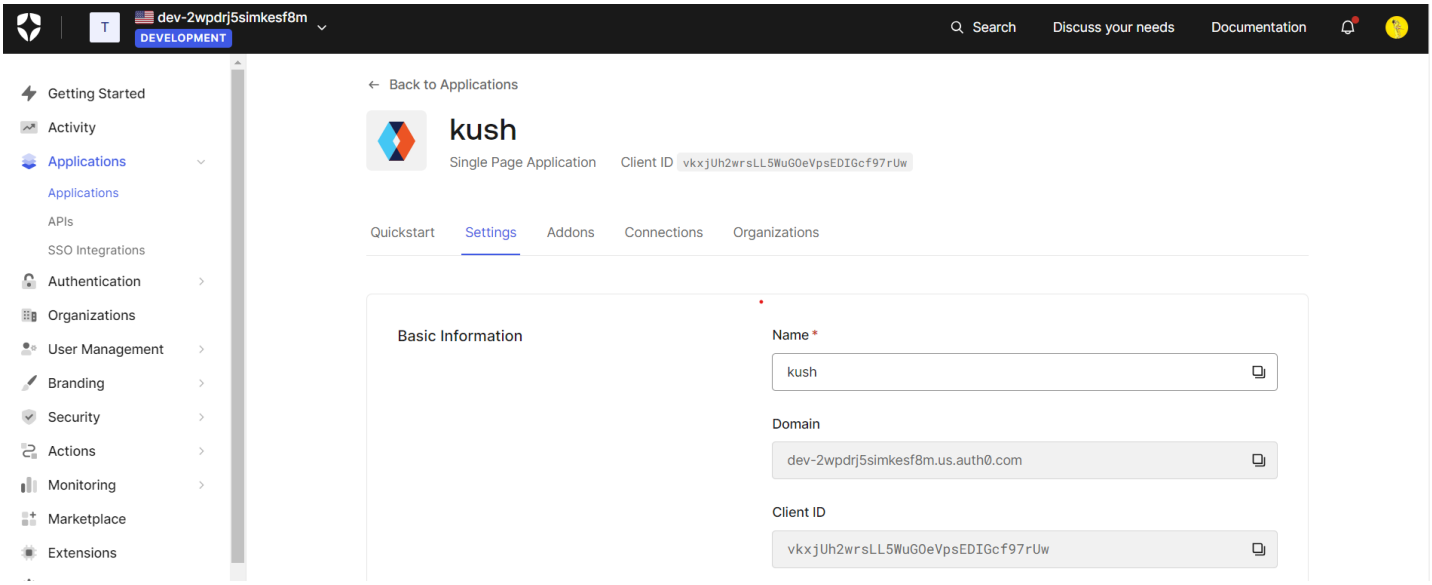
  - ### Efficient Registration and Authentication:
    Administrators or auditors can add users directly into the platform, providing their necessary details such as email and name. Upon addition, users automatically receive a welcome email containing their login credentials, including their email and a temporary password.

  - ### Role-Based Access Control:
    Auth0's role management functionalities empower administrators to assign specific roles to users based on their organizational positions. These roles determine the level of access and permissions each user has within the platform, ensuring data security and compliance.

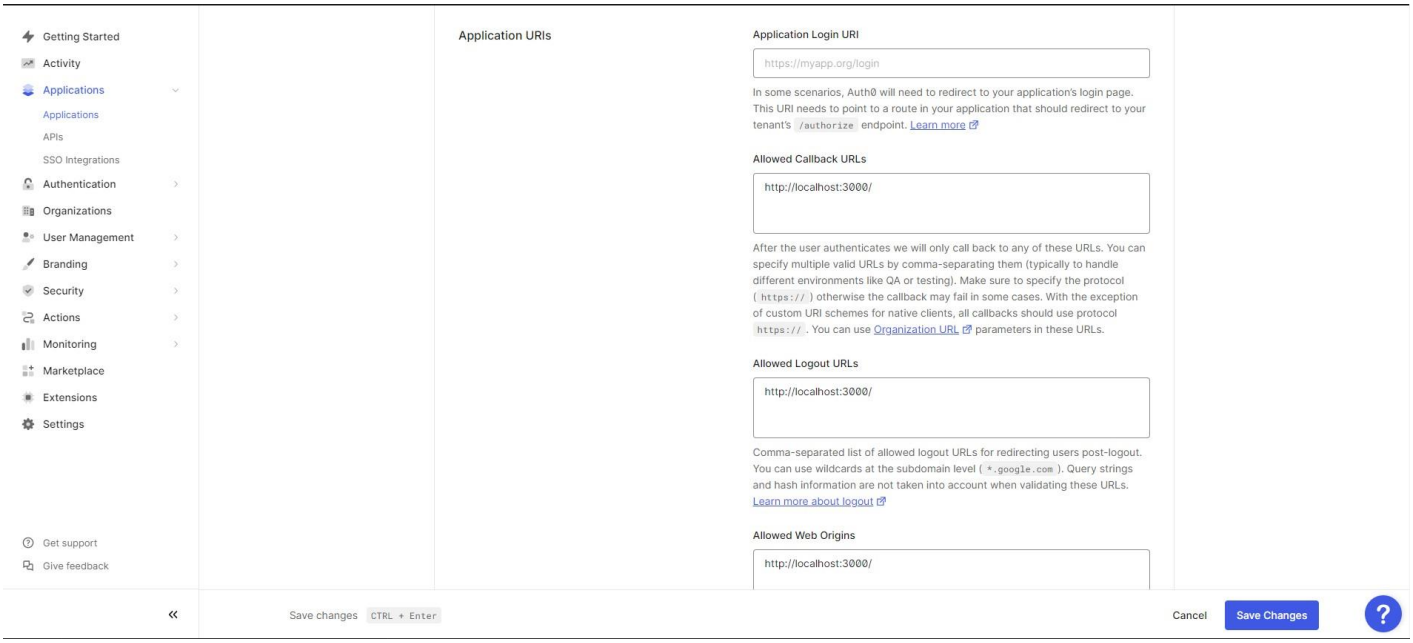
  - ### Password Management and Security:

    Auth0 securely manages user passwords, employing industry-standard encryption techniques to safeguard sensitive information. Users receive their temporary password via email, and they can subsequently reset their password securely if needed, ensuring continuous access to the platform.

  - ### Implementation
    - Created a Single Page Application on Auth0



- Configured required URLs for Application



- **CRUD for Each Section of Project:**

  The 13 Sections CRUD feature empowers users to efficiently manage and manipulate various aspects of the Customer Success Platform. Each section corresponds to a critical component of project management, ranging from project budget to client feedback. Here's an overview of the functionality and role-based access control for each section:

  - **Version History:**

    - **Role Based Access**
      Create/Update/Delete = Admin, Project Manager
      Read = All Users

    - **Endpoints:**
      - **"/versionhistory"**
        - **HTTP Methods** = "GET", "POST"
        - **Status Codes** = 200, 500

- **Project Description:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/projectoverviews"**
      - **HTTP Methods** = "GET", "POST","DELETE","PUT"
      - **Status Codes** = 200, 500,400
- **Scope and stack :**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/scopeandstake"**
      - **HTTP Methods** = "GET", "POST", "DELETE","PUT"
      - **Status Codes** = 200, 500,400
- **Escalation Matrix:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/escalation-matrix"**
      - **HTTP Methods** = "GET", "POST", "DELETE"," PUT"
      - **Status Codes** = 200, 500,400

,4

- **Stakeholders:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Auditor
    Read = All Users
  - **Endpoints:**
    - **"/project/:id/stakeholders"**
      - **HTTP Methods** = "GET", "POST" ,"DELETE","PUT"
      - **Status Code** = 200, 500,400
- **Risk Profiling:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/project/:id/risk-profilling"**
      - **HTTP Methods** = "GET", "POST" ,"DELETE","PUT"
      - **Status Code** = 200, 500,400
- **Phases:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/phasemilestones"**
      - **HTTP Methods** = "GET", "POST","PUT","DELETE"
      - **Status Codes** = 200, 500,400
- **Sprint Wise Details:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/sprintdetails"**
      - **HTTP Methods** == "GET", "POST","PUT","DELETE"
      - **Status Codes** = 200, 500,400

- **Resources:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **“/resources”**
      - **HTTP Methods** = “GET”, “POST” ,”DELETE”,”PUT”
      - **Status Code** = 200, 500,400
- **Client Feedback:**
  - **Role Based Access**
    Create/Update/Delete = Client
    Read = All Users
  - **Endpoints:**
    - **“"/clientfeedbacks”**
      - **HTTP Methods** = “GET”, “POST” ,”DELETE”,”PUT”
      - **Status Code** = 200, 500,400

- **MoM of Client Meetings:**
  - **Role Based Access**
    Create/Update/Delete = Admin, Project Manager
    Read = All Users
  - **Endpoints:**
    - **"/clientmeetingmoms"**
    - **HTTP Methods** = "GET", "POST" ,"DELETE","PUT"
    - **Status Code** = 200, 500,400
- **Audit History:**
  - **Role Based Access**
    Create/Update/Delete = Auditor
    Read = All Users
  - **Endpoints:**
    - **"/audithistory"**
    - **HTTP Methods** = "GET", "POST" ,"DELETE","PUT"
    - **Status Code** = 200, 500,400

• Role Based Management

In our Customer Success Platform, role-based management ensures that users have appropriate access levels tailored to their responsibilities. Below are the roles defined along with their respective permissions:

- **Admin Role:**

- **Full Project Access:** Admins have full access to create, update, read, and delete all projects within the Customer Success Platform.
- **Section Access:** Admins can create, update, read, and delete all sections of each project in the Customer Success Platform.
- **User Management:** Admins can manage users by creating, reading, updating, and deleting all stakeholders' accounts.

- **Auditor Role:**

- **Project Access:** Auditors can create new projects or select existing ones.

- **Project Manager Assignment:** Auditors can assign project managers to projects, facilitating project-specific access for managers.

- **Stakeholder Management:** Auditors can add stakeholders to projects, ensuring their inclusion in the Stakeholders table of the Customer Success Platform.

- **Platform Access:** Auditors can view the Customer Success Platform for all projects.

- **Audit History Comments:** Auditors can add comments to the Audit History table, facilitating documentation and communication within the platform.

○ Project Manager Role:

- **Platform Access:** Project Managers have access to add, edit, and delete existing content within the Customer Success Platform for their assigned project/s.

- **Submission:** Project Managers can save and submit updated Customer Success Platform content for their assigned project/s.

○ Stakeholders/Clients Role:

- **Platform Access:** All stakeholders have view access to the Customer Success Platform for projects assigned to them. This allows them to stay informed and engaged with project updates and information.

- **Email Notification System**

The Email Notification System is designed to keep stakeholders informed about updates and changes within the Customer Success Platform by sending email notifications in real-time. Below is a detailed description of this feature based on the provided requirements:

- **Description**
  The Email Notification System serves as a vital communication channel, ensuring stakeholders are promptly notified about any updates or changes within the platform. This feature is responsible for integrating email notification functionality seamlessly into the platform and developing triggers to send notifications for specific events, such as updates to the Audit History table.

- **Functionality**
  - When updates are made to the Audit History table within the Customer Success Platform, the Email Notification System automatically triggers an email notification to be sent to all stakeholders associated with the project.

  - The email notification contains pertinent information regarding the update, such as the nature of the change, timestamp, and any additional context deemed relevant.

  - Stakeholders receive the email notification in real-time, enabling them to stay informed and engaged with the latest developments within the platform.

- **Implementation**

  - Utilized **Google's gmail** service for sending emails.

  - Endpoints:

On update or create of audit history it is called.

- ## Export as Document

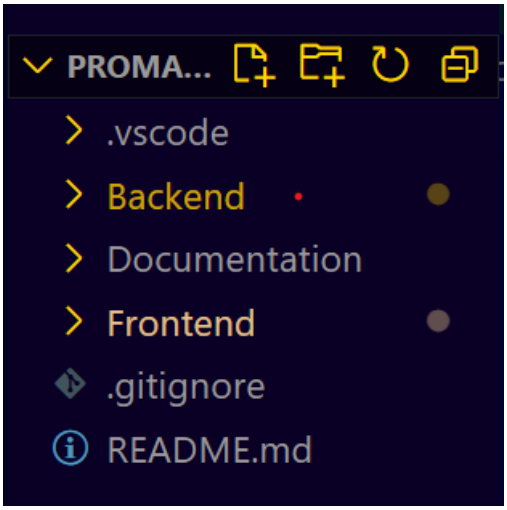
The PDF Document Generation feature enables the creation of comprehensive PDF documents containing all project-related tables and their respective data within the Customer Success Platform. Developed using the html which is embedded in the pdf code it is not done with package:

- ### Functionality:
  - **Inclusion of All Tables:** The PDF document includes all tables present within the Customer Success Platform, ensuring comprehensive coverage of project-related information.
  - **Dynamic Data Population:** Data from each table is dynamically populated into the PDF document, ensuring accuracy and up-to-date information representation.
  - **Customizable Formatting:** Users have the flexibility to customize the formatting and layout of the PDF document, including styling options such as fonts, colors, and table layouts.
  - **Efficient Compilation:** The generation process is efficient and streamlined, allowing users to quickly compile and download PDF documents with just a few clicks.

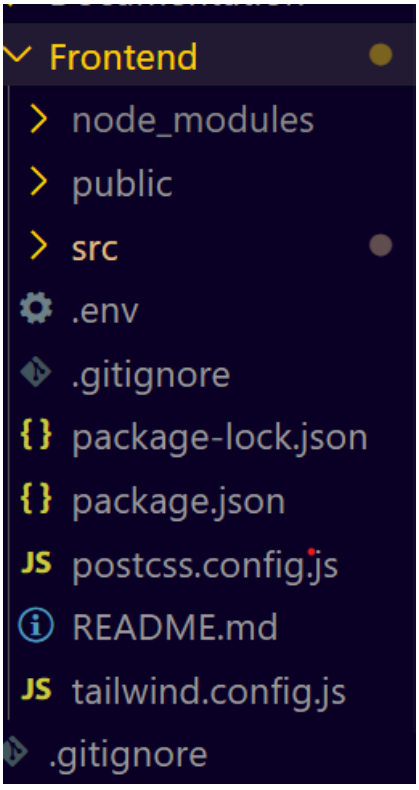
- ### Endpoints:
  - **"/export"**
  - **@GetMapping("/pdf")**
  - HTTP Methods: "GET"
  - Status Codes: 200, 500

- **Folder Structure:**

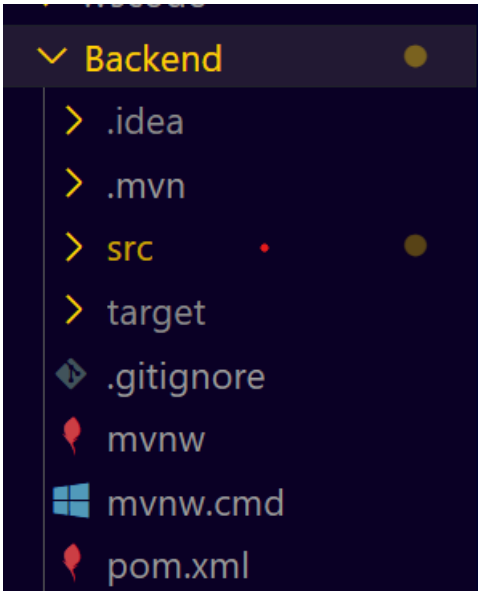
Root Folder Structure



Frontend Folder Structure



Backend Folder Structure



Setup Guide

Follow below steps for setting up the application and using Customer Success Platform:

- **Local Setup**
  - Clone the repository in your local by using below command:

    git clone https://github.com/KushSawlani/Promact_Customer_Success.git
- **Backend Setup:**
  - **Step 1** = Navigate to Backend folder by using below command in terminal:

    cd Backend
  - **Step 2** = Now run below command for installing dependencies required by application:

    mvn spring-boot:run

→Prerequisites:

Before running the backend, ensure you have the following prerequisites installed and configured:

1. **JDK (Java Development Kit): ** Install JDK version new. You can download it from [Oracle JDK] (https://www.oracle.com/java/technologies/javase-jdk11-downloads.html) or [OpenJDK](https://adoptopenjdk.net/).

2. **Maven: ** Install Maven build tool version new You can download it from [Maven Apache] (https://maven.apache.org/download.cgi) and follow the installation instructions.

3. **Environment Variables: ** Ensure that JDK and Maven paths are correctly set in the environment variables and system variables of your operating system.

- Frontend Setup
  - **Step 1** = Now open another terminal and use below command to navigate to Frontend folder (Assuming to be currently in root folder):

    cd Frontend

  - **Step 2** = Now run below command for installing dependencies required by application:

    npm install

  - **Step 4** = Now run below command to start the frontend server:

    npm start

You will be able to see the URL on console, denoting that React server is running successfully. Navigate to that URL in your browser

Now you will be able to use the application.

Credentials for Testing Application

Default datasedded:

- **Admin:**
  - **Email** = admin@gmail.com
  - **Password will have to signup in db as haven't used management api.**

=