



گزارش پروژه کارشناسی  
دانشکده مهندسی  
گروه مهندسی کامپیوتر

“توسعه فرانت‌اند وب سایت مدیریت پروژه”

نویسنده:

ملیکا حسن زاده اسکافی

استاد راهنما:

دکتر محمدجواد ثبوتی

سال تحصیلی 1403-1404

## سپاسگزاری

از خداوند متعال سپاسگزارم که به من توفیق داد تا این پروژه را به پایان برسانم و دانش و مهارت‌های خود را در زمینه توسعه وب افزایش دهم.

از خانواده عزیزم، پدر و مادر گرامی‌ام که همیشه حمایت‌کننده بودند و با صبر و تحمل‌شان، مسیر تحصیل و کارم را هموار کردند، کمال تشکر را دارم. بدون عشق و حمایت آن‌ها، این موفقیت ممکن نبود.

از دوستان خوبم که در لحظات سخت و آسان همراهم بودند و با مشورت‌ها و ایده‌هایشان به من کمک کردند، سپاسگزارم. دوستی شما همیشه منبع الهام برای من بوده است.

در نهایت، از استاد راهنمای محترم، جناب دکتر ثبوتی، که با راهنمایی‌های ارزشمند و دانش عمیق‌شان، این پروژه را هدایت کردند، قدردانی می‌کنم. تجربیات و پیشنهادهای شما نقش کلیدی در تکمیل این کار داشت.

## چکیده

این پروژه کارشناسی به توسعه فرانت‌اند یک وبسایت مدیریت پروژه با استفاده از فریم‌ورک Quasar می‌پردازد. هدف اصلی، ایجاد یک پلتفرم کاربرپسند برای مدیریت پروژه‌ها، تسک‌ها، کاربران و منابع است که امکاناتی مانند داشبورد، مدیریت تسک‌ها، گزارش‌گیری و همکاری تیمی را فراهم می‌کند. پروژه با تمرکز بر رابط کاربری cross-platform طراحی شده و از تکنولوژی‌های مدرن مانند Quasar، Vue.js و ابزارهای کمکی مانند ChatGPT و Grok بهره می‌برد.

در این گزارش، ابتدا اهمیت مدیریت پروژه بررسی می‌شود، سپس به معرفی Quasar و تکنولوژی‌های استفاده‌شده پرداخته می‌شود. در ادامه، توضیحات مفصلي از هر صفحه پروژه ارائه خواهد شد. این وبسایت می‌تواند به سازمان‌ها کمک کند تا بهره‌وری خود را افزایش دهند و پروژه‌ها را به طور کارآمد مدیریت کنند. نتایج نشان می‌دهد که استفاده از Quasar سرعت توسعه را افزایش داده و رابط کاربری باکیفیتی ایجاد کرده است.

پروژه تست‌شده بر روی مرورگرهای مختلف. پیشنهادهای آینده شامل ادغام با بک‌اند واقعی و افزودن ویژگی‌های پیشرفته مانند هوش مصنوعی برای پیش‌بینی تسک‌ها است.

# فهرست مطالب

5	مقدمه
5	اهمیت مدیریت پروژه
5	نقش نرم افزارهای مدیریت پروژه
6	اهداف پروژه
7	معرفی فریم ورک Quasar
8	تکنولوژی های استفاده شده
9	توضیحات صفحات پروژه
9	مستندات صفحه لاگین (UserLogin)
16	مستندات صفحه ثبت نام (SignUp)
19	مستندات صفحه پروفایل کاربر (UserProfile)
22	مستندات صفحه اصلی (homePage)
26	مستندات صفحه جزئیات آیتم ها (itemDetail)
29	مستندات صفحه خطای 404 (ErrorNotFound)
31	ارزیابی و خروجی
41	راهنمای اجرای پروژه
42	برتری های پروژه نسبت به Jira و Trello
43	پیشنهادهای آینده
44	مراجع

## مقدمه

### اهمیت مدیریت پروژه

مدیریت پروژه یکی از جنبه‌های کلیدی در موفقیت سازمان‌ها و کسب‌وکارها است. بدون مدیریت مناسب، پروژه‌ها ممکن است با تأخیر، هزینه‌های اضافی یا شکست مواجه شوند. طبق تعریف، مدیریت پروژه شامل برنامه‌ریزی، اجرا و نظارت بر کارها برای دستیابی به اهداف خاص در زمان و بودجه تعیین‌شده است.

نرم‌افزارهای مدیریت پروژه به تیم‌ها کمک می‌کنند تا کارها را سازماندهی کنند، پیشرفت را پیگیری نمایند و همکاری را بهبود بخشند. برای مثال، این نرم‌افزارها می‌توانند منابع را بهینه کنند، که منجر به افزایش بهره‌وری می‌شود. یکی از مزایای اصلی، بهبود ارتباطات تیمی است؛ جایی که اعضا می‌توانند به راحتی تسک‌ها را به اشتراک بگذارند و بازخورد دهند.

در دنیای امروز، با افزایش پروژه‌های پیچیده، استفاده از ابزارهای دیجیتال ضروری است. این ابزارها نه تنها زمان را صرفه‌جویی می‌کنند، بلکه ریسک‌ها را کاهش می‌دهند و کیفیت خروجی را افزایش می‌دهند. برای نمونه، در صنایع IT، مدیریت پروژه می‌تواند نرخ موفقیت پروژه‌ها را تا ۳۰٪ افزایش دهد. همچنین، در دوران پاندمی، ابزارهای آنلاین مانند این وبسایت، امکان کار از راه دور را تسهیل کرده‌اند.

در ادامه، به جزئیات بیشتری می‌پردازیم. مدیریت پروژه شامل مراحل مختلفی مانند شروع، برنامه‌ریزی، اجرا، نظارت و خاتمه است. هر مرحله نیاز به ابزارهایی دارد که نرم‌افزارهای مدرن فراهم می‌کنند. بدون این ابزارها، تیم‌ها ممکن است با مشکلات مانند عدم هماهنگی یا از دست دادن داده‌ها مواجه شوند.

### نقش نرم‌افزارهای مدیریت پروژه

نرم‌افزارهای مدیریت پروژه مانند Asana، Trello یا Jira، مزایای زیادی دارند. آن‌ها به کاربران اجازه می‌دهند تسک‌ها را ایجاد، اولویت‌بندی و پیگیری کنند. یکی از کلیدی‌ترین مزایا، مدیریت منابع است.

در این پروژه، تمرکز بر فرانت‌اند است، اما قابلیت ادغام با بک‌اند برای ذخیره داده‌ها وجود دارد. این نرم‌افزارها می‌توانند گزارش‌های واقعی‌زمان تولید کنند، که برای تصمیم‌گیری مدیران مفید است. همچنین، امنیت داده‌ها و دسترسی کنترل‌شده از ویژگی‌های مهم هستند.

برای طولانی کردن این بخش، می‌توان به مثال‌های واقعی اشاره کرد. مثلاً در یک شرکت نرم‌افزاری، استفاده از چنین ابزاری می‌تواند زمان تکمیل پروژه را ۲۰٪ کاهش دهد. مطالعات نشان می‌دهد که ۷۷٪ سازمان‌های موفق از نرم‌افزارهای مدیریت پروژه استفاده می‌کنند.

## اهداف پروژه

هدف اصلی، ایجاد یک وب‌سایت مدیریت پروژه با Quasar است. اهداف فرعی شامل: بهبود رابط کاربری، پشتیبانی از دستگاه‌های مختلف، و ادغام ابزارهای AI برای کمک به توسعه.

# معرفی فریم‌ورک Quasar

## تاریخچه و توسعه Quasar

Quasar یک فریم‌ورک open-source بر پایه Vue.js است که برای ساخت اپلیکیشن‌های cross-platform طراحی شده. توسعه آن از سال ۲۰۱۵ آغاز شد و توسط Razvan Stoenescu ایجاد گردید. Quasar امکان ساخت وبسایت، PWA، اپ موبایل و دسکتاپ را با یک کدبیس فراهم می‌کند.

## ویژگی‌های کلیدی Quasar

Quasar شامل کامپوننت‌های آماده مانند QForm، QTable، QDialog و QForm است که توسعه را سرعت می‌بخشد. همچنین، ابزار CLI قدرتمندی دارد که پروژه را سریع راه‌اندازی می‌کند. Quasar از تم‌های تاریک/روشن پشتیبانی می‌کند و با پلاگین‌های خارجی سازگار است.

## مزایا و معایب

مزایا: سرعت بالا، کدبیس واحد، جامعه قوی.

معایب: یادگیری اولیه ممکن است سخت باشد برای تازه‌کاران، و وابستگی به Vue.js.

## مقایسه با دیگر فریم‌ورک‌ها

در مقایسه با Quasar cross-platform، Vuetify، بیشتر دارد. نسبت به React Native، ساده‌تر برای وب است.

## تکنولوژی‌های استفاده‌شده

### Grok

Grok یک مدل هوش مصنوعی ساخته‌شده توسط xAI است که برای کمک به توسعه‌دهندگان استفاده شد. Grok در تولید کد، توضیح مفاهیم و حتی نوشتن این گزارش کمک کرد. به عنوان یک دستیار هوشمند، سؤالات فنی را پاسخ می‌دهد و ایده‌های جدیدی پیشنهاد می‌کند.

### ChatGPT

ChatGPT، مدل زبانی OpenAI، برای brainstorming ایده‌ها، نوشتن کد اولیه و رفع اشکال استفاده شد. این ابزار در تولید محتوای متنی و توضیحات کمک‌کننده بود.

### Cursor

Cursor یک ویرایشگر کد AI است که بر پایه VS Code ساخته شده و با هوش مصنوعی کد را پیشنهاد می‌دهد. در این پروژه، برای نوشتن کامپوننت‌های Vue و رفع خطاها استفاده شد. ویژگی‌هایی مانند تکمیل خودکار کد و ویرایش با پرامپت، توسعه را سرعت بخشید.

### مستندات Quasar

مستندات رسمی (quasar.dev) منبع اصلی برای یادگیری کامپوننت‌ها و بهترین شیوه‌ها بود. مثال‌های آماده در آن، پیاده‌سازی را آسان کرد.

### Canva

Canva برای طراحی لوگو، آیکون‌ها و عناصر گرافیکی استفاده شد. این ابزار آنلاین، بدون نیاز به مهارت پیشرفته، طرح‌های حرفه‌ای ایجاد می‌کند.



## Color Space

Color Space یک ابزار آنلاین برای تولید پالت‌های رنگ و گرادیان‌ها است. در این پروژه، برای انتخاب رنگ‌های تم وبسایت استفاده شد تا رابط کاربری جذاب‌تر شود. این ابزار کمک می‌کند تا رنگ‌ها را بر اساس اصول طراحی انتخاب کنیم.

## VS Code

VS Code یک محیط برنامه نویسی ساده است که استفاده از آن بسیار ساده است. در این پروژه برای کد نویسی راحت‌تر و اجرای بهتر از این محیط استفاده شده است.

## Git

از Git برای کنترل پروژه و ثبت تاریخچه و روند پروژ استفاده شده است.

## Chrome/Firefox/Edge

از مرورگرهای مختلفی برای تست و اجرای برنامه استفاده شد که مهم‌ترین‌های آنها نام برده شده‌اند.

## توضیحات صفحات پروژه

### مستندات صفحه لاگین (UserLogin)

#### هدف و عملکرد صفحه

صفحه لاگین وبسایت مدیریت پروژه **PLANOVA** به کاربران امکان می‌دهد تا با وارد کردن نام کاربری یا ایمیل و رمز عبور، وارد حساب کاربری خود شوند. این صفحه نقطه ورود اصلی به سیستم است و برای احراز هویت کاربران طراحی شده است. هدف اصلی، ارائه یک رابط کاربری ساده، امن و کاربر پسند است. این صفحه به دلیل نقش محوری‌اش در تجربه کاربری، با طراحی جذاب و کاربر پسند، اعتماد کاربران را جلب می‌کند و امنیت را از طریق اعتبارسنجی قوی تضمین می‌کند.

## کامپوننت‌های استفاده‌شده

صفحه لاگین از چندین کامپوننت **Quasar** استفاده می‌کند که توسعه سریع و رابط کاربری منسجم را ممکن کرده‌اند:

- **q-page**: به عنوان پایه صفحه برای مدیریت layout کلی.
- **q-card**: برای ایجاد یک کارت ورود با سایه و گوشه‌های گرد (border-radius: 16px).
- **q-card-section**: برای سازماندهی محتوا داخل کارت.
- **q-form**: برای مدیریت فرم ورود و ارسال داده‌ها.
- **q-input**: برای فیلدهای ورودی نام کاربری/ایمیل و رمز عبور با قابلیت اعتبارسنجی و نمایش آیکون.
- **q-icon**: برای آیکون نمایش/مخفی کردن رمز عبور.
- **q-checkbox**: برای گزینه "Remember Me".
- **q-btn**: برای دکمه ورود با حالت loading.
- **q-banner**: برای نمایش پیام‌های خطا (مثل اشتباه بودن رمز عبور).
- **router-link**: برای لینک به صفحات ثبت‌نام و بازیابی رمز عبور.

## توضیحات کد

### ساختار قالب (Template)

صفحه از یک ساختار تمیز و متمرکز استفاده می‌کند که در مرکز صفحه قرار گرفته است.

توضیحات قالب:

- پس‌زمینه: از تصویر پس‌زمینه (/back3.png) برای جذابیت بصری استفاده شده است. این تصویر با background-size: cover تنظیم شده تا در تمام دستگاه‌ها به خوبی نمایش داده شود.
- نوار ناوبری: یک نوار ثابت (fixed) در بالای صفحه با لوگو و نام برند (PLANOVA) که با box-shadow جلوه‌ای مدرن دارد.
- کارت ورود: کارت با عرض ثابت (420px) و حداکثر عرض vw95 برای سازگاری با صفحه‌نمایش‌های کوچک.
- فرم: شامل دو فیلد ورودی (نام کاربری/ایمیل و رمز عبور)، چک‌باکس "Remember Me" و دکمه ورود. استفاده از autofocus روی فیلد نام کاربری تجربه کاربری را بهبود می‌بخشد.
- لینک‌ها: لینک‌های "Forgot Password" و "Sign up" برای ناوبری سریع.

## منطق جاوااسکریپت (Script)

بخش منطقی صفحه با استفاده از **Composition API** در **Vue 3** نوشته شده است.

توضیحات منطقی:

- استفاده از **ref** برای مدیریت نام کاربری، رمز عبور، نمایش رمز، وضعیت لودینگ، خطاها و گزینه **"Remember Me"**.
- اعتبارسنجی: بررسی اینکه نام کاربری/ایمیل وارد شده و رمز عبور حداقل ۸ کاراکتر باشد.
- شبیه‌سازی **API**: از **localStorage** برای شبیه‌سازی دیتابیس کاربران (**kanbanUsers**) استفاده شده است. در صورت موفقیت، اطلاعات کاربر در **userProfile** ذخیره و توکنی برای احراز هویت تولید می‌شود.
- **Remember Me**: اگر فعال باشد، اطلاعات ورود در **localStorage** ذخیره می‌شود و در بارگذاری بعدی به صورت خودکار پر می‌شود.
- **Auto-dismiss**: استفاده از **composable useAutoDismiss** برای محو شدن خودکار پیام‌های خطا.

## استایل (Style)

استایل‌ها با استفاده از **scoped CSS** نوشته شده‌اند تا از تداخل جلوگیری شود.

توضیحات استایل:

- پس‌زمینه: تصویر پس‌زمینه تمام صفحه را پوشش می‌دهد و با **min-height: 100vh** اطمینان حاصل می‌شود که کل صفحه را پر کند.
- نوار ناوبری: طراحی ثابت با سایه و گوشه‌های گرد برای جلوه مدرن.
- فیلدهای ورودی: ارتفاع فیلدها (**px48**) برای راحتی کاربر تنظیم شده است.
- لینک‌ها: رنگ آبی (**#114992**) و افکت **hover** برای جذابیت بصری.

## تعاملات کاربر

- ورود اطلاعات: کاربر می‌تواند نام کاربری یا ایمیل و رمز عبور را وارد کند. فیلد نام کاربری به صورت خودکار فوکوس می‌شود.
- نمایش/مخفی کردن رمز عبور: آیکون چشم در فیلد رمز عبور امکان تغییر نوع ورودی (**text/password**) را فراهم می‌کند.

- چک باکس **Remember Me**: اگر فعال باشد، اطلاعات ورود ذخیره می شود.
- دکمه ورود: با کلیک یا **Enter**، فرم ارسال می شود و وضعیت لودینگ نمایش داده می شود.
- لینک ها: کاربر می تواند به صفحات ثبت نام یا بازیابی رمز عبور هدایت شود.
- پیام های خطا: در صورت خطا (مثل رمز اشتباه)، پیام در یک **q-banner** نمایش داده می شود و پس از چند ثانیه محو می شود.

### ارتباط با API (شبیه سازی)

- در حال حاضر، صفحه از **localStorage** برای شبیه سازی دیتابیس کاربران استفاده می کند:
- داده های کاربران در **kanbanUsers** ذخیره می شوند.
  - پس از ورود موفق، اطلاعات پروفایل در **userProfile** و یک توکن در **authToken** ذخیره می شود.
  - شبیه سازی با تأخیر ۱ ثانیه ای (**setTimeout**) برای تقلید درخواست **API**.

### چالش ها و راه حل ها

1. چالش: اعتبارسنجی بلادرنگ  
مشکل: نمایش خطاها به صورت پویا بدون رفرش صفحه.  
راه حل: استفاده از **computed** برای اعتبارسنجی رمز عبور و توابع **clearUsernameError** و **clearPasswordError** برای پاک کردن خطاها هنگام تایپ.
2. چالش: مدیریت **Remember Me**  
مشکل: ذخیره امن اطلاعات ورود بدون به خطر انداختن امنیت.  
راه حل: استفاده از **localStorage** برای ذخیره موقت و حذف آن در صورت غیرفعال بودن گزینه.

## مستندات صفحه فراموشی رمز عبور (**ForgetPass**)

### هدف و عملکرد صفحه

صفحه فراموشی رمز عبور در وبسایت مدیریت پروژه **PLANOVA** به کاربرانی که رمز عبور خود را فراموش کرده اند امکان می دهد تا با وارد کردن نام کاربری یا ایمیل، کد تأیید دریافت کنند و رمز عبور جدیدی تنظیم کنند. این صفحه برای بهبود تجربه کاربری و افزایش امنیت طراحی شده است.

این صفحه با طراحی جذاب و فرآیند چندمرحله‌ای، اعتماد کاربر را جلب کرده و از طریق اعتبارسنجی‌های متعدد، امنیت را تضمین می‌کند.

### کامپوننت‌های استفاده‌شده

صفحه فراموشی رمز عبور از مجموعه‌ای از کامپوننت‌های **Quasar** بهره می‌برد که توسعه سریع و رابط کاربری یکپارچه را ممکن کرده‌اند:

- **q-page**: پایه صفحه برای مدیریت layout کلی.
- **q-card**: کارت اصلی با عرض ثابت (px420) و گوشه‌های گرد برای فرم بازنشانی.
- **q-card-section**: سازماندهی محتوا داخل کارت.
- **q-form**: مدیریت فرم برای تأیید نام کاربری/ایمیل و کد تأیید.
- **q-input**: فیلدهای ورودی برای نام کاربری/ایمیل، کد تأیید، رمز جدید و تأیید رمز.
- **q-btn**: دکمه‌های تأیید، اعتبارسنجی کد و ذخیره رمز جدید.
- **q-icon**: آیکون نمایش/مخفی کردن رمز عبور و آیکون قفل در دیالوگ.
- **q-banner**: نمایش پیام‌های خطا (مثل نام کاربری نامعتبر یا کد اشتباه).
- **q-dialog**: دیالوگ برای تنظیم رمز جدید با فرم جداگانه.
- **q-separator**: جداکننده برای بهبود خوانایی.
- **router-link**: لینک بازگشت به صفحه لاگین.

### توضیحات کد

#### ساختار قالب (Template)

صفحه از یک ساختار متمرکز با کارت و دیالوگ استفاده می‌کند.

توضیحات قالب:

- پس‌زمینه: مشابه صفحه لاگین، از تصویر `back3.png` با `background-size: cover` استفاده شده است.
- نوار ناوبری: نوار ثابت با لوگو و نام برند (PLANOVA) برای یکپارچگی بصری.
- کارت اصلی: کارت با عرض `px420` و حداکثر عرض `vw95`

- فرم تأیید: شامل فیلد ورودی نام کاربری/ایمیل و دکمه تأیید. پس از تأیید، فیلد کد تأیید ظاهر می‌شود.
- دیالوگ رمز جدید: دیالوگ با دو فیلد برای رمز جدید و تأیید آن، همراه با دکمه‌های **Save** و **Cancel**.
- لینک بازگشت: لینک به صفحه لاگین برای راحتی کاربر.

### منطق جاوااسکریپت (Script)

بخش منطقی با **Composition API** نوشته شده و فرآیند چند مرحله‌ای را مدیریت می‌کند. توضیحات منطقی:

- استفاده از **ref** برای مدیریت متغیرهایی مثل `usernameOrEmail`، `verificationCode`، `newPassword` و غیره.
- اعتبارسنجی:
  - بررسی وجود نام کاربری/ایمیل با **regex** برای ایمیل و جستجو در `kanbanUsers`.
  - بررسی طول رمز (حداقل ۸ کاراکتر) و عدم تطابق با رمز قبلی.
  - بررسی تطابق رمز جدید و تأیید آن.
- شبیه‌سازی ارسال کد: کد تأیید به صورت ثابت (1234) برای تست استفاده شده است. نسخه اصلی کد تصادفی تولید می‌کند.
- بازنشانی رمز: پس از تأیید کد، دیالوگ باز شده و رمز جدید در `kanbanUsers` ذخیره می‌شود.
- **Auto-dismiss**: پیام‌های خطا با `useAutoDismiss` به صورت خودکار محو می‌شوند.

### استایل (Style)

استایل‌ها مشابه صفحه لاگین هستند و از **scoped CSS** استفاده شده است. توضیحات استایل:

- پس‌زمینه و نوار ناوبری: مشابه صفحه لاگین برای یکپارچگی بصری.
- دکمه‌ها: دکمه‌های گرد با ارتفاع **px48** برای ظاهر مدرن.

### تعاملات کاربر

- ورود نام کاربری/ایمیل: کاربر نام کاربری یا ایمیل را وارد می‌کند و با کلیک روی "Verify" یا **Enter**، فرآیند تأیید شروع می‌شود.
- ورود کد تأیید: پس از تأیید، فیلد کد ۴ رقمی ظاهر شده و کاربر می‌تواند کد را وارد کند.

- اعتبارسنجی کد: دکمه "Validate Code" تنها با ورود ۴ رقم فعال می‌شود.
- دیالوگ رمز جدید: پس از تأیید کد، دیالوگ با دو فیلد (رمز جدید و تأیید) باز می‌شود. کاربر می‌تواند رمز را مشاهده/مخفی کند.
- دکمه‌های **Cancel/Save**: امکان انصراف یا ذخیره رمز جدید.
- پیام‌های خطا: نمایش خطاها (مثل کد اشتباه) در q-banner با محو شدن خودکار.
- بازگشت به لاگین: لینک "Back to Login" برای ناوبری سریع.

#### ارتباط با API (شبیه‌سازی)

- مانند صفحه لاگین، از `localStorage` برای شبیه‌سازی دیتابیس استفاده شده است:
- تأیید کاربر: جستجو در `kanbanUsers` برای یافتن نام کاربری/ایمیل.
  - ارسال کد: کد ثابت 1234 برای تست (نسخه اصلی کد تصادفی تولید می‌کند).
  - بازنشانی رمز: به‌روزرسانی رمز در `kanbanUsers` و ذخیره در `localStorage`.

#### چالش‌ها و راه‌حل‌ها

1. چالش: مدیریت فرآیند چندمرحله‌ای  
مشکل: انتقال بین مراحل (تأیید، کد، رمز جدید) بدون سردرگمی کاربر.  
راه‌حل: استفاده از متغیرهای `emailVerificationSent` و `showPasswordDialog` برای کنترل نمایش بخش‌ها.
2. چالش: شبیه‌سازی ارسال کد  
مشکل: عدم وجود سرور برای ارسال کد واقعی.  
راه‌حل: استفاده از کد ثابت 1234 و کامنت کردن تولید کد تصادفی برای تست.
3. چالش: اعتبارسنجی رمز جدید  
مشکل: جلوگیری از استفاده از رمز قبلی یا رمز ضعیف.  
راه‌حل: استفاده از `computed` برای بررسی طول و تطابق با رمز قبلی.

## مستندات صفحه ثبت نام (SignUp)

### هدف و عملکرد صفحه

صفحه ثبت نام در وبسایت مدیریت پروژه **PLANOVA** به کاربران جدید امکان می دهد تا حساب کاربری ایجاد کنند. این صفحه برای ثبت اطلاعات اولیه کاربر (ایمیل، نام کاربری و رمز عبور) و تأیید ایمیل از طریق کد تأیید طراحی شده است. با استفاده از فریم ورک **Quasar**، این صفحه رابط کاربری کاربرپسندی ارائه می دهد که با دستگاه های مختلف (موبایل، تبلت و دسکتاپ) سازگار است. این صفحه با طراحی جذاب، فرآیند ثبت نام چندمرحله ای و اعتبارسنجی های قوی، تجربه ای امن و روان برای کاربران فراهم می کند.

### کامپوننت های استفاده شده

صفحه ثبت نام از مجموعه ای از کامپوننت های **Quasar** بهره می برد که توسعه سریع و رابط کاربری منسجم را تضمین می کنند:

- **q-page**: پایه صفحه برای مدیریت layout کلی.
- **q-card**: کارت اصلی با عرض ثابت (420px) و حداکثر عرض vw90 برای فرم ثبت نام.
- **q-card-section**: سازماندهی محتوا داخل کارت.
- **q-form**: مدیریت فرم برای جمع آوری اطلاعات و ارسال داده ها.
- **q-input**: فیلدهای ورودی برای ایمیل، کد تأیید، نام کاربری، رمز عبور و تأیید رمز.
- **q-btn**: دکمه های ارسال کد تأیید، تأیید ایمیل و ثبت نام.
- **q-icon**: آیکون نمایش/مخفی کردن رمز عبور.
- **q-banner**: نمایش پیام های خطا یا موفقیت (مثل تأیید ایمیل).
- **q-separator**: جداکننده برای بهبود خوانایی.
- **router-link**: لینک بازگشت به صفحه لاگین.



## توضیحات کد

### ساختار قالب (Template)

صفحه از یک کارت متمرکز با فرم چندمرحله‌ای استفاده می‌کند.

توضیحات قالب:

- پس‌زمینه: استفاده از تصویر /back3.png با background-size: cover برای جذابیت بصری.
- نوار ناوبری: نوار ثابت با لوگو و نام برند (PLANOVA) برای یکپارچگی با سایر صفحات.
- کارت ثبت‌نام: کارت با عرض px420، حداکثر عرض vw90 و حداکثر ارتفاع calc(100vh - 140px) برای جلوگیری از سرریز در دستگاه‌های کوچک.
- فرم: شامل فیلدهای ایمیل، کد تأیید، نام کاربری، رمز عبور و تأیید رمز. فیلد ایمیل با autofocus تجربه کاربری را بهبود می‌بخشد.
- بنرها: نمایش پیام‌های خطا یا موفقیت (مثل تأیید ایمیل) با q-banner.
- لینک بازگشت: لینک به صفحه لاگین برای ناوبری سریع.

### منطق جاوااسکریپت (Script)

بخش منطقی با **Composition API** نوشته شده و فرآیند چندمرحله‌ای ثبت‌نام را مدیریت می‌کند.

توضیحات منطقی:

- استفاده از ref برای مدیریت ایمیل، نام کاربری، رمز عبور، کد تأیید و غیره. متغیرهای Touched برای کنترل نمایش خطاها پس از تعامل کاربر.
- اعتبارسنجی:
  - ایمیل: بررسی فرمت با regex.
  - رمز عبور: حداقل ۸ کاراکتر.
  - نام کاربری و تأیید رمز: بررسی خالی نبودن و تطابق.
- شبیه‌سازی ارسال کد: کد ثابت 1234 برای تست (نسخه اصلی کد تصادفی تولید می‌کند).
- ثبت‌نام: بررسی عدم تکرار ایمیل یا نام کاربری در kanbanUsers. ذخیره پروفایل کاربر و توکن.
- **Auto-dismiss**: پیام‌های خطا و موفقیت با useAutoDismiss محو می‌شوند.

## استایل (Style)

استایل‌ها مشابه صفحات لاگین و فراموشی رمز عبور هستند.

توضیحات استایل:

- پس‌زمینه و نوار ناوبری: یکپارچگی بصری با سایر صفحات.
- کارت: تنظیم max-height برای جلوگیری از سرریز در دستگاه‌های کوچک.
- دکمه‌ها: دکمه‌های گرد با ارتفاع 48px.

## تعاملات کاربر

- ورود ایمیل: کاربر ایمیل را وارد کرده و با کلیک روی "Send Code" کد تأیید دریافت می‌کند.
- ورود کد تأیید: فیلد کد ۴ رقمی با دکمه "Verify Email" که فقط با ۴ رقم فعال می‌شود.
- ورود اطلاعات: نام کاربری، رمز عبور و تأیید رمز با امکان نمایش/مخفی کردن.
- ثبت‌نام: دکمه "Sign Up" فقط پس از تأیید ایمیل فعال می‌شود.
- پیام‌ها: نمایش پیام موفقیت (تأیید ایمیل) یا خطا (مثل ایمیل تکراری) در q-banner.
- بازگشت به لاگین: لینک برای کاربرانی که حساب دارند.

## ارتباط با API (شبیه‌سازی)

از localStorage برای شبیه‌سازی دیتابیس استفاده شده است:

- تأیید ایمیل: کد ثابت 1234 برای تست.
- ثبت‌نام: بررسی عدم تکرار ایمیل یا نام کاربری در kanbanUsers و ذخیره اطلاعات کاربر.
- ذخیره پروفایل: ایجاد userProfile با فیلدهای خالی و توکن.

## چالش‌ها و راه‌حل‌ها

1. چالش: فرآیند چندمرحله‌ای

مشکل: مدیریت مراحل (ایمیل، کد، ثبت‌نام) بدون سردرگمی.

راه‌حل: استفاده از متغیرهای emailVerificationSent و emailVerified برای کنترل نمایش.

2. چالش: اعتبارسنجی بلادرنگ

مشکل: نمایش خطاها فقط پس از تعامل کاربر.

راه حل: استفاده از متغیرهای Touched و computed برای مدیریت خطاها.

3. چالش: شبیه سازی API

مشکل: عدم وجود بک اند.

راه حل: استفاده از کد ثابت و localStorage.

## مستندات صفحه پروفایل کاربر (UserProfile)

### هدف و عملکرد صفحه

صفحه پروفایل کاربر در وبسایت مدیریت پروژه **PLANOVA** به کاربران امکان می دهد تا اطلاعات شخصی خود را مشاهده، ویرایش و مدیریت کنند. این صفحه برای نمایش و به روز رسانی جزئیات پروفایل مانند نام کاربری، نام، نام خانوادگی، تاریخ تولد، ایمیل، شغل، بیو و تصویر پروفایل طراحی شده است. همچنین، قابلیت هایی مانند تغییر رمز عبور، حذف حساب و بازگشت به صفحه اصلی را فراهم می کند. این صفحه با طراحی با امنیت بالا (مانند اعتبارسنجی رمز و جلوگیری از تکرار) و تعاملات کاربر پسند، تجربه ای کامل برای مدیریت پروفایل فراهم می کند. در صورتی که توکن وجود نداشته باشد، کاربر به صفحه لاگین هدایت می شود.

### کامپوننت های استفاده شده

صفحه پروفایل از کامپوننت های گسترده ای از **Quasar** استفاده می کند تا رابط کاربری پویا ایجاد شود:

- **q-page**: پایه صفحه برای layout کلی.
- **q-card**: کارت اصلی با کلاس **constrained-card** برای محدودیت اندازه.
- **q-card-section**: بخش های ثابت و اسکرول شدنی برای سازماندهی محتوا.
- **q-form**: مدیریت فرم ویرایش پروفایل.
- **q-input**: فیلدهای ورودی برای نام کاربری، نام، نام خانوادگی، تاریخ تولد، ایمیل، شغل، بیو، رمز فعلی، رمز جدید و تأیید رمز (با ماسک و اعتبارسنجی).
- **q-avatar**: نمایش تصویر پروفایل یا آیکون پیش فرض.

- **q-btn**: دکمه‌های ویرایش، ذخیره، لغو، حذف، تغییر رمز، آپلود تصویر و غیره.
- **q-icon**: آیکون‌های نمایش/مخفی کردن رمز، تقویم، هشدار و غیره.
- **q-banner**: نمایش پیام‌های خطا یا موفقیت.
- **q-dialog**: دیالوگ‌های حذف حساب و تغییر رمز عبور.
- **q-separator**: جداکننده‌ها برای بهبود خوانایی.
- **q-popup-proxy** و **q-date**: انتخاب‌گر تاریخ برای فیلد تاریخ تولد.
- **router-link**: لینک بازگشت به صفحه اصلی (/home).

### توضیحات کد

کد صفحه طولانی است و بخشی از آن ترونکیت شده، اما بر اساس محتوای موجود، توضیحات را ارائه می‌دهم.

### ساختار قالب (Template)

صفحه از یک کارت با بخش ثابت (تصویر و پیام‌ها) و بخش اسکرول‌شدنی (فرم) استفاده می‌کند.  
توضیحات قالب:

- پس‌زمینه و ناوبری: مشابه صفحات قبلی برای یکپارچگی.
- کارت: با کلاس **constrained-card** برای محدودیت اندازه و جلوگیری از سرریز.
- بخش ثابت: شامل تصویر پروفایل و دکمه‌های آپلود/حذف در حالت ویرایش.
- بخش اسکرول‌شدنی: فرم با فیلدهای ورودی که در حالت غیرویرایش **readonly** هستند.
- دیالوگ‌ها: برای حذف حساب (با هشدار) و تغییر رمز (با اعتبارسنجی رمز فعلی).

### منطق جاوااسکریپت (Script)

اسکریپت با **Composition API** نوشته شده و مدیریت حالت ویرایش، ذخیره و حذف را بر عهده دارد.  
توضیحات منطقی:

- حالت‌ها: برای پروفایل، حالت ویرایش، پیام‌ها، دیالوگ‌ها و غیره.
- اعتبارسنجی: بررسی نام کاربری (عدم تکرار)، ایمیل (فرمت)، تاریخ تولد (فرمت YYYY-MM-DD) و رمز (طول حداقل ۸).

- ذخیره پروفایل: به روزرسانی kanbanUsers و userProfile در localStorage. در صورت تغییر نام کاربری، انتقال آیتم‌ها و دعوت‌نامه‌ها.
- حذف حساب: حذف از kanbanUsers، پاک کردن داده‌های مرتبط و خروج.
- تغییر رمز: دیالوگ با تأیید رمز فعلی و ذخیره رمز جدید.
- آپلود تصویر: بررسی نوع (image/\*) و اندازه (>MB5)، تبدیل به Data URL.
- onMounted: بارگذاری پروفایل و بررسی احراز هویت.

### استایل (Style)

توضیحات استایل:

- کارت محدود: برای جلوگیری از سرریز در دستگاه‌های کوچک.
- بخش ثابت sticky: برای نگه داشتن تصویر در بالا هنگام اسکرول.
- فیلدهای یکنواخت: تنظیم ارتفاع و حاشیه برای زیبایی.

### تعاملات کاربر

- نمایش پروفایل: بارگذاری خودکار اطلاعات.
- ویرایش: کلیک روی "Edit Profile" برای فعال کردن فیلدها و دکمه‌ها.
- تغییر اطلاعات: وارد کردن داده‌ها با اعتبارسنجی بلادرنگ (مثل تاریخ تولد با تقویم).
- تصویر پروفایل: در حالت ویرایش، آپلود یا حذف تصویر.
- تغییر رمز: دیالوگ با فیلدهای رمز فعلی/جدید و دکمه Verify/Save.
- ذخیره/لغو: ذخیره تغییرات یا بازگشت به حالت قبلی.
- حذف حساب: دیالوگ تأیید برای حذف دائمی.
- پیام‌ها: بنرهای خطا/موفقیت با محو شدن خودکار.
- بازگشت به خانه: لینک "Back to Home".

### ارتباط با API (شبیه‌سازی)

از localStorage برای شبیه‌سازی:

- بارگذاری پروفایل: از userProfile و kanbanUsers.
- ذخیره: به روزرسانی kanbanUsers و userProfile، انتقال داده‌ها در تغییر نام.

- حذف: پاک کردن داده‌ها از `kanbanUsers`، آیتم‌ها و دعوت‌نامه‌ها.
- تغییر رمز: به‌روزرسانی رمز در `kanbanUsers`.

#### چالش‌ها و راه‌حل‌ها

1. چالش: مدیریت حالت ویرایش  
مشکل: حفظ داده‌های قبلی در لغو.  
راه‌حل: بک‌آپ با `backupProfile` و `watch` روی `editMode`.
2. چالش: انتقال داده‌ها در تغییر نام  
مشکل: حفظ آیتم‌ها و دعوت‌نامه‌ها.  
راه‌حل: جستجو و به‌روزرسانی `kanbanItems` و `kanbanInvitations`.
3. چالش: آپلود تصویر  
مشکل: بررسی اندازه و نوع.  
راه‌حل: استفاده از `FileReader` و محدودیت‌ها.
4. چالش: حذف حساب  
مشکل: پاک کردن داده‌های مرتبط.  
راه‌حل: فیلتر کردن از تمام کاربران و پاک کردن `localStorage`.

### مستندات صفحه اصلی (homePage)

#### هدف و عملکرد صفحه

صفحه اصلی (خانه) در وبسایت مدیریت پروژه **PLANOVA** به عنوان داشبورد اصلی عمل می‌کند و کاربران را قادر می‌سازد تا پروژه‌ها، تسک‌ها و آیتم‌های خود را در یک برد کانبان مدیریت کنند. این صفحه برای نمایش، ویرایش، جابجایی، اضافه کردن، حذف و به اشتراک‌گذاری آیتم‌ها در سه وضعیت (`Backlog`, `In Progress`, `Done`) طراحی شده است. همچنین، ویژگی‌هایی مانند جستجو، سورت (بر اساس اولویت یا تاریخ)، نمایش پیشرفت، مدیریت دعوت‌نامه‌ها و پیام‌ها، پروفایل و لاگ‌اوت را فراهم می‌کند.

این صفحه، تعاملات درگاند دروپ و شبیه‌سازی ذخیره‌سازی محلی، یک محیط مدیریت پروژه کارآمد ارائه می‌دهد. آیتم‌ها می‌توانند ساب‌آیتم داشته باشند و تغییرات در آیتم‌های به اشتراک گذاشته‌شده، به طور هم‌زمان به‌روزرسانی می‌شوند.

### کامپوننت‌های استفاده‌شده

صفحه اصلی از کامپوننت‌های گسترده **Quasar** برای ایجاد یک رابط کاربری پویا و **interactive** استفاده می‌کند:

- **q-page**: پایه صفحه با کلاس **fixed-page** برای مدیریت ارتفاع.
- **q-input**: فیلدهای جستجو، فرم اضافه کردن (عنوان، توضیح، مهلت و غیره).
- **q-btn**: دکمه‌های لاگ‌اوت، پروفایل، اضافه کردن آیتم، ذخیره، حذف، پذیرش/رد دعوت و غیره.
- **q-toggle**: تگل سورت (اولویت/تاریخ).
- **q-card**: کارت‌های ستون‌های کانبان و فرم‌ها.
- **q-card-section**: بخش‌های کارت برای عنوان و محتوا.
- **q-separator**: جداکننده‌ها در کارت‌ها.
- **q-banner**: نمایش پیام‌های خطا، دعوت‌نامه‌ها و پیام‌ها.
- **q-badge**: نمایش تعداد آیتم‌ها، پیام‌ها و دعوت‌نامه‌ها.
- **q-icon**: آیکون‌های جستجو، نوتیفیکیشن، حذف و غیره.
- **q-circular-progress**: نمایش درصد پیشرفت.
- **q-form**: مدیریت فرم اضافه کردن آیتم.
- **q-select**: انتخاب نوع آیتم (Project/Task)، اولویت و نقش به اشتراک‌گذاری.
- **q-date** و **q-popup-proxy**: انتخاب‌گر تاریخ برای مهلت.
- **q-dialog**: دیالوگ‌های جزئیات آیتم، ویرایش، پیام‌ها و دعوت‌نامه‌ها.
- **q-list** و **q-item**: لیست کاربران برای به اشتراک‌گذاری، پیام‌ها و دعوت‌نامه‌ها.
- **router-link**: لینک به پروفایل (در کد به صورت **click@** استفاده شده).

### توضیحات کد

کد صفحه بسیار طولانی است و بخشی ترونکیت شده، اما بر اساس محتوای موجود، نکات کلیدی را توصیف می‌کنم.

## ساختار قالب (Template)

صفحه از **layout** ثابت با نوار بالا، برد کانبان و پنل راست استفاده می‌کند

توضیحات قالب:

- نوار بالا: تاریخ، جستجو و اقدامات (پروفایل، لاگ‌اوت).
- برد کانبان: سه ستون با کارت‌ها برای آیتم‌ها، پشتیبانی از درگ‌اندروپ و فیلتر جستجو.
- پنل راست: پیشرفت دایره‌ای، بنر دعوت‌نامه‌ها، دکمه اضافه کردن و فرم (با فیلدها و لیست به اشتراک گذاری).
- دیالوگ‌ها: برای پیام‌ها، دعوت‌نامه‌ها (پذیرش/رد) و جزئیات آیتم (ویرایش، ساب‌آیتم‌ها، کامنت‌ها).

## منطق جاوااسکریپت (Script)

اسکریپت مدیریت آیتم‌ها، ذخیره‌سازی محلی و تعاملات را بر عهده دارد.

توضیحات منطقی:

- حالت‌ها: برای آیتم‌ها، پیام‌ها، دعوت‌نامه‌ها، فرم، دیالوگ‌ها و غیره.
- بارگذاری/ذخیره: استفاده از `localStorage` برای `kanbanItems_{$username}` و `kanbanInvitations_{$username}` و `kanbanMessages_{$username}`.
- اضافه کردن/ویرایش: تولید ID منحصر به فرد، به اشتراک گذاری با دعوت‌نامه و نقش‌ها.
- درگ‌اندروپ: مدیریت با `draggedItem` و بررسی اجازه ویرایش (`canEdit`).
- به اشتراک گذاری: ارسال دعوت‌نامه، پذیرش/رد و به‌روزرسانی آیتم‌ها در کاربران مرتبط.
- `onMounted` و `watch`: بارگذاری اولیه، نظارت بر تغییرات `localStorage` و به‌روزرسانی لیست.

## استایل (Style)

توضیحات استایل:

- برد و کارت‌ها: گرد و شفاف برای زیبایی.
- رنگ اولویت: رنگ‌بندی کارت‌ها بر اساس اولویت.

## تعاملات کاربر

- جستجو: فیلتر آیتم‌ها با ورودی متن.
- سورت: تگل برای تغییر معیار سورت.



- درگاندروپ: جابجایی آیتم‌ها بین ستون‌ها با بررسی اجازه.
- اضافه کردن آیتم: باز کردن فرم، انتخاب نوع/اولویت، اضافه کاربران برای اشتراک.
- مشاهده/ویرایش آیتم: کلیک روی آیتم برای دیالوگ جزئیات (ویرایش، ساب‌آیتم، کامنت).
- حذف: دکمه حذف با تأیید اجازه.
- دعوت‌نامه‌ها: نمایش تعداد، دیالوگ برای پذیرش/رد.
- پیام‌ها: دیالوگ برای مشاهده و حذف پیام‌ها.
- پیشرفت: نمایش درصد تکمیل.
- پروفایل/لاگ‌اوت: دکمه‌ها برای ناوبری.

### ارتباط با API (شبیه‌سازی)

از `localStorage` برای ذخیره‌سازی محلی:

- آیتم‌ها: ذخیره در `kanbanItems_${username}`
  - دعوت‌نامه‌ها: در `kanbanInvitations_${username}`.
  - پیام‌ها: در `kanbanMessages_${username}`.
  - به‌روزرسانی اشتراکی: هنگام تغییر، به‌روزرسانی در کاربران مرتبط و مدیریت تعارض‌ها.
- در آینده: اتصال به `API` واقعی برای هم‌زمانی سرور.

### چالش‌ها و راه‌حل‌ها

1. چالش: مدیریت اشتراک‌گذاری  
مشکل: هم‌زمانی تغییرات در کاربران متعدد.  
راه‌حل: `updateSharedItems` برای به‌روزرسانی و مدیریت حذف/تعارض.
2. چالش: درگ‌اندروپ با شرایط  
مشکل: جلوگیری از حرکت به `Done` بدون تکمیل ساب‌آیتم‌ها.  
راه‌حل: بررسی در `handleDrop`.

3. چالش: ذخیره محلی پویا

مشکل: نظارت بر تغییرات `localStorage`.

راه حل: `watch` روی کلیدها برای بروزرسانی.

4. چالش: اجازه ویرایش

مشکل: محدود کردن اقدامات به صاحب/ادمین.

راه حل: تابع `canEdit` بر اساس نقش.

## مستندات صفحه جزئیات آیتم‌ها (itemDetail)

### هدف و عملکرد صفحه

صفحه جزئیات آیتم‌ها در وبسایت مدیریت پروژه **PLANOVA** برای نمایش و مدیریت جزئیات یک آیتم خاص (پروژه یا تسک) طراحی شده است. این صفحه بر اساس ID آیتم (از `route.params.id`) بارگذاری می‌شود و امکان مشاهده، ویرایش، مدیریت ساب‌آیتم‌ها، کامنت‌ها، گزارش‌ها، جلسات، تکالیف و تاریخچه تغییرات را فراهم می‌کند. همچنین، نقش کاربر (مانند Admin یا View-only) را بررسی کرده و دسترسی‌ها را محدود می‌کند.

### کامپوننت‌های استفاده‌شده

صفحه از کامپوننت‌های **Quasar** برای رابط کاربری `interactive` استفاده می‌کند:

- **q-page**: پایه صفحه با کلاس `fixed-page`.
- **q-input**: فیلدهای جستجو، عنوان، مهلت، کامنت، گزارش، جلسه، تکلیف و نوت.
- **q-btn**: دکمه‌های ویرایش، ذخیره، لغو، حذف، اضافه نوت/کامنت/گزارش و غیره.
- **q-toggle**: تگل سورت (اولویت/تاریخ).
- **q-card**: کارت‌های جزئیات، نوت، تاریخچه و دیالوگ‌ها.
- **q-separator**: جداکننده‌ها.
- **q-banner**: پیام‌های خطا.
- **q-icon**: آیکون‌های ویرایش، حذف، تقویم و غیره.

- **q-select**: انتخاب وضعیت، اولویت و دسته‌بندی.
- **q-date** و **q-popup-proxy**: انتخاب‌گر تاریخ برای مهلت.
- **q-dialog**: دیالوگ نوت، ویرایش کامنت/گزارش و غیره.
- **q-list** و **q-item**: لیست کاربران برای اشتراک، کامنت‌ها، گزارش‌ها و تاریخچه.
- **router-link**: لینک به پروفایل/خانه (با @click).

## توضیحات کد

کد طولانی و ترونکیت شده است، اما نکات کلیدی را توصیف می‌کنم.

## ساختار قالب (Template)

صفحه از **layout** مشابه خانه با جزئیات آیتم و ساب‌آیتم‌ها استفاده می‌کند

توضیحات قالب:

- نوار بالا: نمایش عنوان آیتم و نقش کاربر.
- جستجو و اقدامات: جستجو ساب‌آیتم‌ها، پروفایل و لاگ‌اوت.
- نوت: نمایش **inline** با دکمه‌های ویرایش/حذف (اگر اجازه باشد).
- زنجیره والد: نمایش مسیر والد تا آیتم فعلی.
- جزئیات آیتم: فیلدهای ویرایشی با غیرفعال کردن بر اساس نقش.
- ساب‌آیتم‌ها: ستون‌های کانبان با درگ‌اندروپ و فیلتر.
- پنل راست: بخش‌های کامنت، گزارش، جلسه، تکلیف و تاریخچه با اضافه/ویرایش (بر اساس نقش).
- دیالوگ‌ها: برای نوت، ویرایش کامنت/گزارش و غیره.

## منطق جاوااسکریپت (Script)

مدیریت بارگذاری، ویرایش و ذخیره آیتم.

توضیحات منطقی:

- حالت‌ها: برای آیتم، جستجو، ویرایش، دیالوگ‌ها و غیره.
- بارگذاری: استخراج آیتم و زنجیره والد از **localStorage**.

- ویرایش: فعال/غیرفعال بر اساس canEdit
- ساب آیتم‌ها: سورت با sortedSubitems و فیلتر جستجو.
- نوت/کامنت/گزارش: اضافه/ویرایش/حذف با بررسی نقش.
- ذخیره: بروزرسانی localStorage و کاربران اشتراکی.
- onMounted: بارگذاری اولیه.

## استایل (Style)

توضیحات استایل:

- گرد و شفاف: برای کارت‌ها و فیلدها.
- رنگ اولویت: برای کارت ساب آیتم‌ها.

## تعاملات کاربر

- بارگذاری: نمایش لودینگ تا بارگذاری آیتم.
  - ویرایش جزئیات: دکمه Edit/Save با فیلدهای فعال (اگر اجازه باشد).
  - نوت: نمایش inline با ویرایش/حذف (دیالوگ برای اضافه/ویرایش).
  - ساب آیتم‌ها: درگ اند دروپ، جستجو، سورت، اضافه (فرم)، ویرایش (دیالوگ)، حذف.+
  - به اشتراک گذاری و دادن دسترسی بر اساس نقش کاربر
- **Admin**: میتواند گزارش و Meeting ایجاد کند و اگر یک آیتم به او اختصاص یافت، میتواند برای آن زیرآیتم ایجاد کند و برای آیتم‌هایی که ایجاد کرده، مانند Owner رفتار کند با این تفاوت که فقط میتواند آیتم را با کاربری با نقش Observer به اشتراک بگذارد.
- **Observer**: فقط مشاهده می‌کند، نمی‌تواند ویرایش یا حذف کند، اما می‌تواند کامنت اضافه کند.
- **Owner**: دسترسی کامل دارد. (تنها کسی است که تاریخچه همه ویرایشات گزارش و توضیحات Meeting ها و کامنت ها را دارد و میتواند به کامنت ها پاسخ بدهد.
- کامنت/گزارش/جلسه/تکلیف: اضافه با فیلد، ویرایش/حذف (با تاریخچه).
  - پیام خطا: برای عدم دسترسی یا خطاها.
  - ناوبری: بازگشت به خانه، پروفایل، لاگ‌اوت.

## ارتباط با API (شبیه‌سازی)

از localStorage برای ذخیره:

- آیتم: در {kanbanItems\_{\$username}.
- به‌روزرسانی اشتراکی: با updateSharedItems.
- ذخیره تغییرات: در کاربران مرتبط.

## چالش‌ها و راه‌حل‌ها

1. چالش: مدیریت نقش‌ها

مشکل: محدود کردن اقدامات.

راه‌حل: computed مانند canEdit برای بررسی Admin/View-only/Owner.

2. چالش: سورت و فیلتر

مشکل: پویایی لیست ساب‌آیتم.

راه‌حل: تابع sortedSubitems با watch.

## مستندات صفحه خطای 404 (ErrorNotFound)

### هدف و عملکرد صفحه

صفحه خطای در وبسایت مدیریت پروژه **PLANOVA** برای نمایش پیام خطا در زمانی که کاربر به مسیری دسترسی پیدا کند که وجود ندارد یا پیدا نمی‌شود، طراحی شده است. این صفحه یک رابط کاربری ساده و مینیمال ارائه می‌دهد که کاربر را از وضعیت خطا آگاه می‌کند و با حفظ طراحی یکپارچه با سایر صفحات اپلیکیشن، تجربه‌ای منسجم فراهم می‌کند.

### کامپوننت‌های استفاده‌شده

صفحه از حداقل کامپوننت‌های **Quasar** برای نمایش خطا استفاده می‌کند:

- **q-page**: پایه صفحه با کلاس login-bg برای پس‌زمینه و چیدمان مرکزی.
- **div**: برای نمایش لوگو، نوار ناوبری و پیام خطا.

هیچ کامپوننت تعاملی مانند `q-input`، `q-btn` یا `q-dialog` استفاده نشده است، زیرا هدف صرفاً نمایش خطاست.

## توضیحات کد

### ساختار قالب (Template)

صفحه از یک ساختار ساده با پیام متنی بزرگ و نوار ناوبری استفاده می‌کند

توضیحات قالب:

- نوار ناوبری: مشابه سایر صفحات با لوگو و نام برند (PLANOVA) برای یکپارچگی.
- پیام خطا: نمایش عدد "404" با فونت بزرگ (vh30) و پیام "Oops. Nothing here..." با استایل `2text-h` و شفافیت 0.9.
- چیدمان مرکزی: استفاده از `flex flex-center` برای قرار گرفتن پیام در وسط صفحه.
- پس‌زمینه: تصویر `back3.png` با `background-size: cover` برای هماهنگی با سایر صفحات.

### منطق جاوااسکریپت (Script)

این صفحه هیچ منطق جاوااسکریپتی ندارد

توضیحات منطقی:

- عدم نیاز به اسکریپت: این صفحه صرفاً نمایشی است و نیازی به مدیریت حالت، API یا تعاملات ندارد.
- سادگی: عدم استفاده از `ref`، `computed` یا `onMounted` به دلیل ماهیت استاتیک صفحه.

### استایل (Style)

توضیحات استایل:

- پس‌زمینه: استفاده از تصویر پس‌زمینه مشابه سایر صفحات برای یکپارچگی بصری.
- نوار ناوبری: ثابت در بالا با سایه و گردی گوشه‌ها.
- لوگو و عنوان: تنظیم اندازه و فاصله برای برندینگ.

## تعاملات کاربر

- نمایش پیام خطا: کاربر به صورت بصری از عدم وجود صفحه مطلع می‌شود.
- بدون تعامل: هیچ دکمه یا فرمی برای تعامل وجود ندارد.
- ناوبری غیرمستقیم: کاربر می‌تواند با استفاده از نوار آدرس یا بازگشت مرورگر به صفحات دیگر برود.

## ارتباط با API

این صفحه هیچ ارتباطی با API (حتی شبیه‌سازی شده) ندارد، زیرا صرفاً یک صفحه استاتیک برای نمایش خطاست.

## نقش‌ها و اجازه‌ها

- بدون محدودیت نقش: این صفحه برای همه کاربران (اعم از مهمان، کاربر وارد شده، Admin یا View-only) قابل دسترسی است، زیرا فقط خطا را نمایش می‌دهد.
- بدون نیاز به احراز هویت: نیازی به بررسی توکن یا نقش نیست، زیرا هیچ داده حساسی نمایش داده نمی‌شود.

## چالش‌ها و راه‌حل‌ها

### 1. چالش: وضوح پیام

مشکل: اطمینان از خوانایی پیام خطا در دستگاه‌های مختلف.

راه‌حل: استفاده از فونت بزرگ (vh30 برای "404") و متن واضح با رنگ سفید و شفافیت.

## ارزیابی و خروجی

برای مستندسازی پروژه، یک سناریوی پیچیده طراحی شده است که همه امکانات پروژه را پوشش می‌دهد. این سناریو از ابتدا (ثبت‌نام کاربر جدید) شروع می‌شود و به تدریج به امکانات پیشرفته مانند اشتراک‌گذاری با نقش‌های مختلف، مدیریت ساب‌آیتم‌ها، ایجاد جلسات، گزارش‌ها، کامنت‌ها و غیره می‌رسد. سناریو بر اساس نقش‌های کاربر (Owner: صاحب آیتم با دسترسی کامل؛ Admin: نقش ویرایشی؛ Observer/View-only: نقش مشاهده‌ای با محدودیت) و دسترسی‌های مختلف پیش می‌رود. نتیجه مطلوب در هر مرحله مشخص شده است.

این سناریو فرض می‌کند سه کاربر وجود دارد: **UserA** (صاحب اصلی) و **UserB** (کاربر دعوت‌شده با نقش Admin) و **UserC** (دعوت‌شده با نقش Observer). پروژه از `localStorage` برای ذخیره‌سازی استفاده می‌کند، بنابراین داده‌ها بین سشن‌ها حفظ می‌شوند. سناریو گام‌به‌گام است و همه صفحات و امکانات را بررسی می‌کند.

## گام 1: ثبت‌نام کاربر جدید (Sign Up)

### ● اقدامات:

- کاربر به صفحه ثبت‌نام (`signup/`) می‌رود.
- ایمیل معتبر (مثل `usera@example.com`) وارد می‌کند و روی "Send Code" کلیک می‌کند. کد تأیید (شبیه‌سازی‌شده: 1234) در کنسول نمایش داده می‌شود.
- کد را وارد کرده و "Verify Email" را کلیک می‌کند تا ایمیل تأیید شود.
- نام کاربری (`UserA`)، رمز عبور (حداقل 8 کاراکتر، مثل `123password`) و تأیید رمز را وارد می‌کند.
- روی "Sign Up" کلیک می‌کند.

### ● اتفاقات داخلی:

- اعتبارسنجی: بررسی فرمت ایمیل، طول رمز، عدم تکرار نام کاربری/ایمیل در `kanbanUsers`.
- ذخیره: ایجاد رکورد در `kanbanUsers` و `userProfile` در `localStorage`، تولید `authToken`.

### ● نتیجه مطلوب:

کاربر به صفحه لاگین هدایت می‌شود. پیام "Sign up successfully" نمایش داده می‌شود. حالا `UserA` ثبت شده و می‌تواند لاگین کند.  
برای بقیه کاربرها نیز همین گونه ثبت نام انجام می‌شود.



## گام 2: لاگین (Login)

### ● اقدامات:

- کاربر به صفحه لاگین (/login) می‌رود.
- نام کاربری (UserA) یا ایمیل و رمز عبور (123password) را وارد می‌کند.
- گزینه "Remember Me" را فعال می‌کند (اختیاری).
- روی "Login" کلیک می‌کند.

### ● اتفاقات داخلی:

- اعتبارسنجی: جستجو در kanbanUsers برای تطابق رمز.
- ذخیره: بارگذاری userProfile و authToken در localStorage.

### ● نتیجه مطلوب:

کاربر به صفحه اصلی (/home) هدایت می‌شود. پیام "Login successful (demo only)" نمایش داده می‌شود. اگر "Remember Me" فعال باشد، اطلاعات ورود ذخیره می‌شود.

## گام 3: فراموشی رمز عبور (Forget Password)

### ● اقدامات:

- کاربر به صفحه فراموشی رمز (/forgetPass) می‌رود.
- نام کاربری (UserA) یا ایمیل را وارد کرده و "Verify" را کلیک می‌کند.
- کد تأیید (شبهه‌سازی شده: 1234) را وارد کرده و "Validate Code" را کلیک می‌کند.
- در دیالوگ، رمز جدید (مثل 456newpass) و تأیید را وارد کرده و "Save" را کلیک می‌کند.

- اتفاقات داخلی:

- اعتبارسنجی: بررسی وجود کاربر در kanbanUsers، کد معتبر، رمز جدید حداقل 8 کاراکتر و متفاوت از قبلی.
- ذخیره: بروزرسانی رمز در kanbanUsers.

- نتیجه مطلوب:

رمز تغییر کرده و کاربر به لاگین هدایت می‌شود. پیام "Password reset successful (demo only)" نمایش داده می‌شود. حالا کاربر با رمز جدید لاگین می‌کند.

#### گام 4: ویرایش پروفایل کاربر (Profile)

- اقدامات (پس از لاگین مجدد با رمز جدید):

- به صفحه پروفایل (/profile) می‌رود.
- روی "Edit Profile" کلیک می‌کند.
- نام کاربری را به UserA\_Updated تغییر می‌دهد، ایمیل، تاریخ تولد، شغل و بیه را ویرایش می‌کند.
- تصویر پروفایل آپلود می‌کند (فایل PNG کمتر از MB5).
- روی "Change Password" کلیک کرده، رمز فعلی را وارد می‌کند، رمز جدید را وارد کرده و ذخیره می‌کند.
- روی "Save" کلیک می‌کند.

- اتفاقات داخلی:

- اعتبارسنجی: بررسی تکراری نبودن نام کاربری جدید، فرمت ایمیل، تاریخ تولد.
- ذخیره: بروزرسانی userProfile و kanbanUsers در localStorage، انتقال آیتم‌ها/دعوت‌نامه‌ها در تغییر نام.

- نتیجه مطلوب:

پروفایل بروزرسانی شده، پیام "Profile updated successfully" نمایش داده می‌شود. تغییرات در localStorage حفظ می‌شود.

## گام 5: ایجاد آیتم جدید در صفحه اصلی (Home)

- اقدامات:

- به صفحه اصلی (home/) می‌رود.
- در پنل راست، روی "Add Item" کلیک می‌کند.
- نوع "Project"، عنوان "Main Project"، مهلت (تاریخ)، حالت "Backlog" اولویت "High" را وارد می‌کند.
- روی "Save" کلیک می‌کند.
- آیتم جدید در ستون Backlog ظاهر می‌شود.

- اتفاقات داخلی:

- اعتبارسنجی: عنوان الزامی، مهلت معتبر.
- ذخیره: اضافه به kanbanItems\_UserA\_Updated در localStorage، ارسال دعوت‌نامه به UserB در kanbanInvitations\_UserB.

- نتیجه مطلوب:

آیتم ایجاد شده، دعوت‌نامه برای UserB ارسال می‌شود. پیشرفت 0٪ نمایش داده می‌شود.

## گام 6: جابجایی آیتم بین حالات برد (Drag-and-Drop)

- اقدامات:

- آیتم "Main Project" را از Backlog به In Progress درگ می‌کند.

○ سپس به Done درگ می‌کند (اگر ساب‌آیتم Done نشده نداشته باشد، موفق می‌شود).

● اتفاقات داخلی:

- بررسی: برای Done، ساب‌آیتم‌های ناتمام بررسی می‌شود.
- ذخیره: بروزرسانی وضعیت در localStorage و کاربران اشتراکی.

● نتیجه مطلوب:

آیتم جابجا شده، پیشرفت بروزرسانی می‌شود (اگر Done باشد، به 100٪ نزدیک می‌شود).

## گام 7: تغییر سوئیچ اولویت و جستجو

● اقدامات:

- تگل "Sort by" را به اولویت تغییر می‌دهد.
- در جستجو، عنوان یا نوع آیتم را وارد می‌کند.

● اتفاقات داخلی:

- سورت: بر اساس اولویت (High > Medium > Low) یا تاریخ.
- فیلتر: آیتم‌های مطابق نمایش داده می‌شوند.

● نتیجه مطلوب:

لیست مرتب‌شده و فیلتر شده نمایش داده می‌شود.

## گام 8: ویرایش نوت و اطلاعات آیتم

● اقدامات:

- در جزئیات، نوت را ویرایش می‌کند.

- اطلاعات آیتم (عنوان، مهلت) را تغییر می‌دهد.

### ● اتفاقات داخلی:

- اعتبارسنجی: نوت اختیاری، مهلت معتبر.
- ذخیره: بروزرسانی در `localStorage`.

### ● نتیجه مطلوب:

تغییرات اعمال شده، پیام موفقیت نمایش داده می‌شود.

## گام 10: ساخت زیر آیتم (Subitem)

### ● اقدامات:

- در جزئیات آیتم، ساب‌آیتم "1 Sub Task" را اضافه می‌کند.

### ● اتفاقات داخلی:

- اضافه به `subitems` ساب‌آیتم اصلی.
- بررسی نقش برای ویرایش ساب‌آیتم.

### ● نتیجه مطلوب:

ساب‌آیتم ایجاد و جابجا شده، پیشرفت اصلی بروزرسانی می‌شود.

## گام 11: به اشتراک گذاشتن آیتم با نقش‌های متفاوت

### ● اقدامات:

- `UserA` آیتم را با `UserB` با نقش `Admin` به اشتراک می‌گذارد.
- `UserA` آیتم را با `UserC` با نقش `Observer` به اشتراک می‌گذارد.

- UserB و UserC لاگین کرده، دعوت‌نامه را در صفحه اصلی می‌پذیرند.
- UserB میتواند در آیتم گزارش بنویسد، Meeting اضافه کند و گزارش بنویسد.

#### ● اتفاقات داخلی:

- دعوت: ذخیره در kanbanInvitations\_UserB.
- پذیرش: اضافه آیتم به kanbanItems\_UserB با نقش.
- تغییرات: هم‌زمان‌سازی با updateSharedItems.
- نتیجه مطلوب: آیتم اشتراک گذاشته میشود.

### گام 12: اختصاص دادن آیتم (Assign Item)

#### ● اقدامات:

- در جزئیات، آیتم را به UserB اختصاص می‌دهد (assign).

#### ● اتفاقات داخلی:

- ذخیره assign در آیتم.
- پیام به UserB ارسال می‌شود.
- UserB میتواند برای آیتمی که به او اختصاص داده شده زیر آیتم بسازد و برای آیتم‌هایی که می‌سازد تغییر اطلاعات دهد و آنها را با کاربری با نقش Observer به اشتراک بگذارد.

#### ● نتیجه مطلوب :

آیتم اختصاص یافته، UserB در پیام‌ها مطلع می‌شود.

### گام 13: ایجاد Meeting

#### ● اقدامات:

- در پنل راست جزئیات، جلسه جدید ("Project Review") را اضافه می‌کند.

- اتفاقات داخلی:

- اضافه به لیست جلسات آیتم.

- نتیجه مطلوب:

جلسه ثبت شده، در تاریخچه نمایش داده می شود.

## گام 14: ثبت و ویرایش گزارش (Report)

- اقدامات:

- گزارش جدید ("Progress Report") اضافه می کند.
- گزارش را ویرایش کرده و ذخیره می کند.

- اتفاقات داخلی:

- اعتبارسنجی: گزارش الزامی نیست.
- ذخیره با timestamp و editedBy.

- نتیجه مطلوب:

گزارش ثبت/ویرایش شده، تاریخچه بروزرسانی می شود.

## گام 15: ثبت و ویرایش کامنت (Comment)

- اقدامات:

- کامنت جدید ("Need more details") اضافه می کند.
- کامنت را ویرایش کرده و ذخیره می کند.

- اتفاقات داخلی:

- اضافه کردن کامنت به لیست کامنت‌ها
- ویرایش فقط توسط صاحب کامنت.

- نتیجه مطلوب:

کامنت ثبت/ویرایش شده، در لیست نمایش داده می‌شود.

## گام 16: حذف آیتم (Delete Item)

- اقدامات:

- آیتم "Main Project" را حذف می‌کند.

- اتفاقات داخلی:

- حذف از `{kanbanItems_{$username}` و کاربران اشتراکی.
- پاک کردن ساب‌آیتم‌ها و دعوت‌نامه‌ها.

- نتیجه مطلوب:

آیتم حذف شده، صفحه اصلی بروزرسانی می‌شود.

## نتیجه کلی سناریو

- اتفاقات نهایی:

همه امکانات تست شده، داده‌ها ذخیره و هم‌زمان‌سازی شده.



## ● نتیجه مطلوب :

سیستم بدون خطا کار می‌کند، نقش‌ها رعایت شده، داده‌ها در `localStorage` حفظ می‌شود. این سناریو همه امکانات را پوشش می‌دهد.

## راهنمای اجرای پروژه

برای اجرای پروژه مدیریت پروژه که با **Quasar Framework** و **Vue.js** توسعه داده شده است، لازم است مراحل زیر طی شود:

### پیش‌نیازها

قبل از شروع، باید ابزارهای زیر روی سیستم نصب باشند:

**Node.js** (ترجیحاً نسخه LTS)

**npm** یا **yarn** (به همراه Node.js نصب می‌شوند)

یک ویرایشگر کد مانند VS Code

### دریافت کد پروژه

کد پروژه می‌تواند از مخزن گیت (Git) یا فایل فشرده دریافت شود

### نصب وابستگی‌ها

در مسیر اصلی پروژه (محلی که فایل `package.json` قرار دارد) دستور زیر اجرا شود:

```
npm install
```

این دستور تمام کتابخانه‌ها و پکیج‌های مورد نیاز پروژه را نصب می‌کند. مدت زمان این مرحله بسته به سرعت اینترنت و تعداد وابستگی‌ها متفاوت است.

## اجرای پروژه در حالت توسعه

برای اجرای برنامه به صورت محلی و مشاهده آن در مرورگر، از دستور زیر استفاده می‌شود:

```
quasar dev
```

پس از اجرای موفق، در ترمینال آدرسی مانند زیر نمایش داده می‌شود:

```
http://localhost:9000/
```

با باز کردن این آدرس در مرورگر، نسخه‌ی در حال توسعه‌ی برنامه بارگذاری خواهد شد.

## برتری‌های پروژه نسبت به Jira و Trello

### سادگی و شخصی‌سازی

- ابزارهایی مثل Jira خیلی پیچیده هستند و معمولاً برای تیم‌های بزرگ طراحی شده‌اند.
- پروژه من سبک و ساده است و برای کاربران تازه‌کار یا تیم‌های کوچک خیلی مناسب‌تر است.

### رایگان و متن‌باز بودن

- Jira نسخه رایگان محدود دارد.
- Trello هم در نسخه رایگان محدودیت‌هایی دارد (مثل تعداد بردها و پلاگین‌ها).
- پروژه‌ی من می‌تواند کاملاً متن‌باز و رایگان باشد، بدون محدودیت برای تیم‌های کوچک.

## تمرکز بر نیازهای خاص

- Jira بیشتر روی مدیریت پروژه‌های نرم‌افزاری و Agile تمرکز دارد.
  - Trello روی مدیریت وظایف ساده‌تر تمرکز دارد.
  - پروژه‌ی من می‌تواند ترکیبی باشد: هم ساده مثل Trello و هم قابل توسعه به سمت امکانات پیچیده‌تر مثل Jira.
  - می‌توان ادعا کرد این پروژه «بینابینی» طراحی شده و نیاز کاربران داخلی را بهتر پوشش می‌دهد.
- «این پروژه با وجود سادگی، رایگان و بومی بودن، می‌تواند جایگزینی مناسب برای ابزارهای پیچیده و غیر رایگان خارجی باشد، به‌خصوص برای تیم‌های کوچک، دانشجویان و سازمان‌های داخلی.»

## پیشنهادهای آینده

1. افزودن بخش بک‌اند (Backend)
2. قابلیت‌های پیشرفته مدیریت پروژه
  - تقویم و زمان‌بندی برای مدیریت وظایف.
  - گانت چارت (Gantt Chart) برای نمایش زمان‌بندی پروژه‌ها.
  - اعلان‌ها (Notifications) برای یادآوری وظایف.
3. اضافه کردن AI برای کمک به مدیریت پروژه

Vue.js Official Documentation — <https://vuejs.org/>  
Quasar Framework Documentation — <https://quasar.dev/>  
Pinia State Management Documentation — <https://pinia.vuejs.org/>  
MDN Web Docs (JavaScript, HTML, CSS) — <https://developer.mozilla.org/>  
GitHub & Open Source Project Repositories (برای مشاهده‌ی نمونه‌های مشابه).