

مقدمه

اپلیکیشن ما یک پلتفرم آموزشی جامع است که به کاربران کمک می‌کند تا برنامه‌نویسی را به شکلی ساده، جذاب و تعاملی یاد بگیرند. این اپلیکیشن به گونه‌ای طراحی شده است که نیازهای کاربران مبتدی تا حرفه‌ای را پوشش دهد و تجربه‌ای شخصی‌سازی شده برای هر کاربر ایجاد کند.

ویژگی‌های اصلی این اپلیکیشن شامل مسیر یادگیری گام‌به‌گام، چالش‌های روزانه، تمرین‌های متنوع، پروژه‌های عملی، و سیستم پاداش‌دهی مبتنی بر گیمیفیکیشن است. کاربران می‌توانند زبان‌های برنامه‌نویسی مختلفی را انتخاب کنند و با توجه به نیاز و اهداف خود، مسیر یادگیری شخصی‌سازی شده‌ای داشته باشند. طراحی بصری مدرن و دسترسی آسان به تمامی ابزارها و محتواها از دیگر ویژگی‌های برجسته این اپلیکیشن است.

هدف اصلی این اپلیکیشن، ایجاد انگیزه برای یادگیری و ارائه یک مسیر آموزشی موثر و کاربردی است که کاربران بتوانند مهارت‌های برنامه‌نویسی خود را در پروژه‌های واقعی به کار گیرند و پیشرفت خود را به صورت مداوم ارزیابی کنند.

صفحات اپلیکیشن به این شرح است:

- صفحه ورود و ثبت‌نام (Login/Signup Page): امکان ورود یا ثبت‌نام با شماره تماس، نام کاربری و رمز عبور، یا حساب گوگل.
- صفحه آشنایی اولیه: (Introduction) انتخاب زبان برنامه‌نویسی یا دریافت پیشنهاد مناسب بر اساس اهداف و علایق کاربر.
- مسیر یادگیری (Learning Path): نمایش گرافیکی دروس، چالش‌ها و پیشرفت کاربر.
- صفحه آموزش درس (Learning Content): ارائه توضیحات جامع به همراه مثال‌های تعاملی و قابل اجرا.
- صفحه تمرین درس (Lesson Practice): تمرین‌های متنوع برای تثبیت مفاهیم

یادگرفته شده.

۶. صفحه پروفایل کاربر (User Profile): نمایش اطلاعات کاربر، دستاوردها و پیشرفت.

۷. صفحه داشبورد (Dashboard): مدیریت روزانه فعالیت‌ها، مشاهده پیشرفت‌ها و پیشنهادات ویژه.

۸. صفحه چالش‌ها (Challenges Page): شرکت در چالش‌های روزانه، هفتگی و ویژه برای تقویت مهارت‌ها.

۹. صفحه جزئیات چالش (Challenge Details): مشاهده جزئیات چالش‌های انتخاب‌شده و تعامل با سوالات مربوطه.

۱۰. صفحه پروژه‌ها (Projects Page): انتخاب و انجام پروژه‌های عملی برای تقویت یادگیری.

۱۱. صفحه جزئیات پروژه (Project Details): راهنمای مرحله‌به‌مرحله برای انجام پروژه‌ها.

۱۲. صفحه تنظیمات و اهداف یادگیری (Settings & Goals): تنظیم اهداف یادگیری، اعلان‌ها و مدیریت حساب کاربری.

هر یک از این صفحات با طراحی منحصر به فرد و امکانات کاربردی، تجربه‌ای روان و موثر برای کاربران فراهم می‌کنند. این مستند به جزئیات طراحی هر صفحه پرداخته و نقشه‌ای جامع برای پیاده‌سازی اپلیکیشن ارائه می‌دهد.

صفحه ورود و ثبت نام(Login/Signup Page)

هدف صفحه ورود و ثبت نام

این صفحه به کاربران اجازه می‌دهد با استفاده از شماره تماس، نام کاربری و رمز عبور، یا حساب گوگل وارد برنامه شوند یا حساب جدید ایجاد کنند. طراحی این صفحه باید ساده، امن، و سریع باشد.

ساختار و جزئیات صفحه ورود

۱. ورود به حساب(Login)

۱. روش‌های ورود:

- ورود با شماره تماس و رمز عبور.
- ورود با نام کاربری و رمز عبور.
- ورود با حساب گوگل.(Google Account)

۲. فرم ورود:

- فیلدهای ورود:
 - شماره تماس یا نام کاربری.
 - رمز عبور.
- دکمه "ورود" برای تایید اطلاعات.
- لینک "رمز عبور خود را فراموش کرده‌اید؟" برای بازیابی رمز عبور.

۳. ورود با گوگل:

- دکمه بزرگ و مشخص با عنوان "ورود با حساب گوگل".

- اتصال به API های گوگل برای احراز هویت.

۴. پیام‌های خطا:

- پیام‌های خطا برای ورود نادرست (مثلاً: "شماره تماس یا رمز عبور اشتباه است").

- پیام‌های مربوط به فیلدهای خالی (مثلاً: "پر کردن تمام فیلدها الزامی است").

ساختار و جزئیات صفحه ثبت‌نام (Signup)

۲. ایجاد حساب جدید

۱. روش‌های ثبت‌نام:

- ثبت‌نام با شماره تماس.

- ثبت‌نام با نام کاربری و رمز عبور.

- ثبت‌نام با حساب گوگل.

۲. فرم ثبت‌نام:

- فیلدهای ثبت‌نام:

- شماره تماس (یا ایمیل).

- نام کاربری.

- رمز عبور.

- دکمه "ثبت‌نام" برای تایید اطلاعات.

۳. پذیرش شرایط و قوانین:

- چک‌باکس برای تایید مطالعه و پذیرش قوانین و مقررات.

- لینک مستقیم به صفحه "قوانین و مقررات."

۴. پیام‌های خطا و موفقیت:

- پیام‌های خطا برای فیلدهای خالی یا نامعتبر (مثلاً: "شماره تماس نامعتبر است").

- پیام تایید پس از ثبت‌نام موفقیت‌آمیز (مثلاً: "ثبت‌نام شما با موفقیت انجام شد!").

ویژگی‌های امنیتی صفحه ورود و ثبت‌نام

۱. رمزنگاری اطلاعات:

- استفاده از HTTPS برای ارسال داده‌ها.

- رمزنگاری رمز عبور در سمت سرور.

۲. تایید هویت دو مرحله‌ای (اختیاری):

- ارسال کد تایید (OTP) به شماره تماس یا ایمیل.

۳. محدودیت تلاش‌های ورود:

- قفل موقت حساب در صورت تلاش مکرر ناموفق.

طراحی بصری پیشنهادی: (UI)

۱. سادگی و مینیمالیستی:

- طراحی صفحه با تمرکز بر راحتی کاربران.

۲. دکمه‌ها و فیلدهای مشخص:

- استفاده از رنگ‌های متمایز برای دکمه‌ها (مثل آبی یا سبز برای تایید).

۳. آیکون‌ها:

- استفاده از آیکون‌های شماره تماس، رمز عبور، و گوگل برای شفافیت بیشتر.

مزایای این طراحی:

۱. **دسترسی آسان**: کاربران می‌توانند با سه روش مختلف وارد حساب خود شوند.
۲. **امنیت بالا**: رمزنگاری داده‌ها و امکان تایید هویت دو مرحله‌ای.
۳. **تجربه کاربری روان**: طراحی ساده و سریع برای ورود و ثبت‌نام.

جمع‌بندی:

صفحه ورود و ثبت‌نام به گونه‌ای طراحی شده که ساده، امن و سریع باشد. این صفحه تجربه کاربری مناسبی ارائه می‌دهد و با پشتیبانی از روش‌های مختلف ورود، کاربران را به راحتی به برنامه متصل می‌کند.

بخش Introduction (آشنایی اولیه با اپلیکیشن)

ویژگی‌های بخش Introduction

نمای کلی صفحه انتخاب زبان و مشاوره

صفحه Introduction ترکیبی از دو بخش اصلی است: لیست زبان‌های برنامه‌نویسی و گزینه مشاوره برای انتخاب زبان مناسب. این ساختار به کاربران کمک می‌کند که یا زبان مورد نظر خود را مستقیم انتخاب کنند یا با پاسخ به چند سوال ساده، زبان مناسب بر اساس علایق و نیازهایشان پیشنهاد شود.

جزئیات بخش‌های صفحه

۱. لیست زبان‌های برنامه‌نویسی

طراحی کارت‌ها: هر زبان به صورت یک کارت شامل:

نام زبان (Python، JavaScript، ++C و ...).

آیکون مرتبط (مثلاً آیکون Python).

۲. گزینه مشاوره برای انتخاب زبان

نمایش گزینه مشاوره:

یک دکمه بزرگ و برجسته با عنوان "راهنمای انتخاب زبان".

مکان نمایش: بالای لیست زبان‌ها یا پایین صفحه.

پرسش سوالات:

سوالات ساده و هدفمند:

آیا به توسعه وب علاقه دارید؟ (بله / خیر)

آیا به هوش مصنوعی یا داده‌کاوی علاقه دارید؟ (بله / خیر)

آیا به ساخت اپلیکیشن‌های موبایل علاقه دارید؟ (بله / خیر)

آیا تجربه‌ای در برنامه‌نویسی دارید؟ (بله / خیر)

هر سوال به صورت کارت مجزا نمایش داده می‌شود تا کاربر به صورت تعاملی پاسخ دهد.

پیشنهاد زبان:

پس از پاسخ به سوالات، زبان‌های مناسب به همراه توضیح کوتاه پیشنهاد داده می‌شوند.

مثال: اگر کاربر به هوش مصنوعی علاقه دارد، Python و R پیشنهاد می‌شود.

۳. سوالات مربوط به زمان و اهداف یادگیری

پرسش درباره زمان یادگیری:

سوال: "روزانه چقدر زمان می‌توانید به یادگیری اختصاص دهید؟"

گزینه‌ها:

۱۰ دقیقه.

۲۰ دقیقه.

بیشتر از ۳۰ دقیقه.

پرسش درباره هدف یادگیری:

سوال: "هدف اصلی شما چیست؟"

گزینه‌ها:

یادگیری مبانی برنامه‌نویسی.

ارتقاء مهارت‌های کدنویسی.

آماده شدن برای مصاحبه شغلی.

جریان کار پیشنهادی

نمایش صفحه انتخاب زبان:

لیست زبان‌های برنامه‌نویسی در کارت‌های جداگانه نمایش داده می‌شود.

گزینه مشاوره در بالای صفحه یا زیر لیست زبان‌ها قرار می‌گیرد.

انتخاب مستقیم یا مشاوره:

کاربر یا یک زبان را مستقیم انتخاب می‌کند و به مرحله بعد هدایت می‌شود.

یا با کلیک روی گزینه مشاوره، به سوال و پاسخ هدایت می‌شود.

پرسش درباره زمان و هدف یادگیری:

پس از انتخاب زبان یا دریافت پیشنهاد از مشاوره، کاربر زمان و هدف یادگیری خود را مشخص می‌کند.

نمایش زبان‌های پیشنهادی:

در صورت استفاده از مشاوره، زبان‌های مناسب به همراه توضیحات کوتاه نمایش داده می‌شوند.

کاربر یکی از زبان‌های پیشنهادی را انتخاب کرده و مسیر یادگیری برای او شخصی‌سازی می‌شود.

رابط کاربری پیشنهادی

بخش لیست زبان‌ها:

طراحی ساده و جذاب با کارت‌های جداگانه برای هر زبان.

هر کارت شامل:

نام زبان.

آیکون یا لوگوی زبان.

بخش مشاوره:

یک دکمه بزرگ و برجسته برای شروع مشاوره.

سوالات به صورت تعاملی و کارت به کارت نمایش داده می‌شوند.

نمایش زبان‌های پیشنهادی در قالب لیست یا کارت.

بخش زمان و هدف:

سوالات با گزینه‌های چندگانه برای انتخاب سریع.

نمایش پیش‌نمایش مسیر یادگیری بر اساس انتخاب کاربر.

بخش مسیر یادگیری (Learning Path)

ویژگی‌های کلیدی مسیر یادگیری

مسیر یادگیری هسته اصلی تجربه کاربر در اپلیکیشن است که شامل نمایش گرافیکی دروس، پیشرفت کاربر، و دسترسی به آموزش و تمرین‌ها می‌شود. طراحی این بخش باید تعاملی، کاربرپسند، و متناسب با نیاز کاربران مبتدی و حرفه‌ای باشد.

ساختار و جزئیات مسیر یادگیری

۱. نمایش گرافیکی مسیر

نمای مسیر:

مسیر یادگیری به صورت یک گراف عمودی طراحی می‌شود.

گره‌ها (Nodes) نماینده درس‌ها یا چالش‌های مختلف هستند.

گره‌های قفل‌شده و بازشده:

گره‌های بازشده: قابل کلیک و با رنگ روشن (مثلاً سبز).

گره‌های قفل‌شده: با آیکن قفل و رنگ خاکستری مشخص می‌شوند.

کاربر برای باز کردن یک گره جدید باید گره قبلی را تکمیل کند.

نوار پیشرفت:

نمایش پیشرفت کاربر در هر گره با یک نوار درصدی یا ستاره.

پیشرفت کلی مسیر یادگیری نیز در بالای صفحه قابل مشاهده است.

۲. آموزش و تمرین جداگانه

عملکرد گره‌ها:

وقتی کاربر روی یک گره کلیک می‌کند، دو گزینه نمایش داده می‌شود:

آموزش: ورود به بخش توضیحات و مفاهیم مرتبط با درس.

تمرین: ورود مستقیم به سوالات و تمرین‌ها برای مرور و تثبیت مفاهیم.

بخش آموزش:

شامل توضیحات متنی و تصویری درباره مفاهیم درس.

مثال‌های کدنویسی قابل اجرا.

بخش تمرین:

سوالات و تمرین‌ها با تنوع بالا:

پر کردن جای خالی در کد.

نوشتن کد برای حل مسئله.

مرتب‌سازی خطوط کد.

بازخورد فوری برای هر سوال.

۳. سیستم پیشرفت و جوایز

ارز مجازی (Gems):

کاربر با تکمیل درس‌ها و تمرین‌ها ارز مجازی کسب می‌کند.

ارزها برای خرید امکانات ویژه مثل نکات آموزشی یا باز کردن گره‌های خاص قابل استفاده هستند.

مدال‌ها و جوایز:

پس از تکمیل هر سطح، کاربر مدال یا جایزه دریافت می‌کند.

چالش‌های ویژه:

در فواصل مشخص، چالش‌هایی برای مرور مباحث قبلی قرار می‌گیرند که پاداش بیشتری دارند.

۴. قابلیت اضافه کردن کورس جدید

گزینه در بالای صفحه:

دکمه‌ای در بالای مسیر یادگیری برای اضافه کردن زبان برنامه‌نویسی یا کورس جدید.

فرایند:

با کلیک بر روی این دکمه، کاربر به صفحه انتخاب کورس هدایت می‌شود.

پس از انتخاب، مسیر یادگیری جدیدی ایجاد می‌شود.

پیشرفت جداگانه:

پیشرفت هر کورس به صورت مستقل نمایش داده می‌شود و کاربر می‌تواند بین کورس‌ها جابه‌جا شود.

۵. نوار پایین (Navigation Bar)

بخش‌های اصلی:

خانه (Home): مسیر یادگیری.

چالش‌ها: دسترسی به چالش‌های ویژه و رقابتی.

پروژه‌ها: تمرین‌های عملی و پروژه‌های کوچک.

پروفايل: نمایش سطح، مدال‌ها، و امتیازات کاربر.

فروشگاه: خرید ارز مجازی یا امکانات خاص.

عملکرد:

نوار پایین با آیکن‌های مشخص، کاربران را به بخش‌های مختلف هدایت می‌کند.

صفحه فعلی با رنگ یا آیکن متمایز مشخص می‌شود.

مزایای این طراحی

انعطاف‌پذیری:

کاربران می‌توانند انتخاب کنند که ابتدا آموزش ببینند یا مستقیماً به تمرین بروند.

تجربه شخصی‌سازی شده:

مسیر یادگیری بر اساس پیشرفت و نیازهای هر کاربر تنظیم می‌شود.

تعامل بیشتر:

قابلیت اجرای کدها و دریافت بازخورد فوری.

گیمیفیکیشن:

سیستم جوایز و ارز مجازی انگیزه کاربران را برای پیشرفت بیشتر افزایش می‌دهد.

دسترسی سریع:

نوار پایین به کاربران اجازه می‌دهد به سرعت به بخش‌های مختلف برنامه دسترسی پیدا کنند.

بخش آموزش درس (Learning Content)

هدف اصلی:

ارائه توضیحات جامع و قابل فهم درباره موضوع درس.

نمایش مثال‌های کدنویسی مرتبط برای تقویت درک کاربر.

طراحی تعاملی که کاربر بتواند مفاهیم را به راحتی یاد بگیرد.

ساختار و جزئیات صفحه آموزش

۱. بخش بالای صفحه:

عنوان درس:

نمایش عنوان درس (مثلاً: "حلقه‌های تکرار در پایتون").

موقعیت درس در مسیر یادگیری (مثلاً: سطح ۱، بخش ۲).

توضیح کوتاه:

یک پاراگراف مختصر که هدف درس را توضیح می‌دهد.

مثال: "در این درس با حلقه‌های for و while آشنا می‌شوید."

۲. محتوای اصلی درس:

توضیحات مفهومی:

متن‌های آموزشی ساده و قابل فهم.

استفاده از نکات کلیدی (Bullet Points) برای توضیح مفاهیم.

مثال: "حلقه for برای تکرار یک بلوک کد تا زمانی که شرط خاصی برقرار باشد استفاده می‌شود."

مثال‌های کدنویسی:

قطعه کدهای مرتبط با موضوع درس.

قابلیت تعاملی برای اجرای مثال‌ها.

ساختار پیشنهادی:

نمایش کد در یک محیط کدنویسی کوچک.

دکمه "اجرا" برای مشاهده خروجی کد.

مثال:

```
for i in range(5):
```

```
    print(i)
```

نکات مهم:

نمایش نکات کلیدی یا هشدارها (مثلاً: "به تعداد دفعات تکرار در حلقه توجه کنید").

استفاده از جعبه‌های رنگی برای برجسته کردن نکات.

۳. تعامل کاربر با محتوا:

محیط تعاملی:

کاربران می‌توانند کدها را تغییر دهند و اجرا کنند.

خروجی کد بلافاصله نمایش داده شود.

پرسش‌های کوچک:

در خلال آموزش، سوالات کوچک یا چالش‌هایی مطرح شود.

مثال: "چه اتفاقی می‌افتد اگر مقدار range را تغییر دهیم؟"

۴. بخش پایانی:

خلاصه درس:

مرور نکات کلیدی درس.

مثال: "در این درس یاد گرفتید چگونه از حلقه for استفاده کنید."

دکمه انتقال به تمرین:

دکمه "حالا تمرین کن!" که کاربر را به بخش تمرین مربوط به درس هدایت می‌کند.

طراحی بصری پیشنهادی:

بخش‌بندی واضح:

هر بخش (عنوان، توضیحات، مثال‌ها) به صورت جداگانه و با فاصله مناسب نمایش داده شود.

استفاده از رنگ‌ها:

رنگ‌های ملایم برای پس‌زمینه و جعبه‌های برجسته برای نکات مهم.

محیط کدنویسی کوچک:

طراحی ساده با قابلیت اجرا و ویرایش کد.

مزایای این طراحی:

تعامل بالا: کاربران می‌توانند مفاهیم را به صورت عملی یاد بگیرند.

ساختار منظم: محتوا به صورت بخش‌بندی‌شده و منسجم ارائه می‌شود.

قابلیت تمرکز: کاربر بدون حواس‌پرتی روی مفاهیم درس تمرکز می‌کند.

آمادگی برای تمرین: کاربران پس از دیدن آموزش آماده ورود به بخش تمرین هستند.

بخش تمرین درس (Lesson Practice)

هدف اصلی:

کمک به کاربر برای مرور و تثبیت مفاهیم یادگرفته شده.

ارائه تمرین‌های متنوع و تعاملی برای تقویت یادگیری عملی.

ارزیابی مهارت کاربر در استفاده از مفاهیم درس.

ساختار و جزئیات بخش تمرین

۱. بخش بالای صفحه:

عنوان درس:

نمایش عنوان درس مرتبط با تمرین (مثلاً: "تمرین حلقه‌های تکرار").

نمایش موقعیت درس در مسیر یادگیری (مثلاً: سطح ۱، بخش ۲).

وضعیت پیشرفت تمرین:

نوار درصدی برای نشان دادن تعداد تمرین‌های تکمیل‌شده.

تعداد سوالات باقی‌مانده و تعداد پاسخ‌های صحیح.

۲. انواع سوالات تمرینی:

برای تنوع و جذابیت بیشتر، می‌توان از انواع مختلف سوالات استفاده کرد:

پر کردن جای خالی در کد:

قطعه کدی با بخش‌های خالی نمایش داده می‌شود و کاربر باید جای خالی را پر کند.

مثال:

```
____ for i in range(5):
```

```
    print(i)
```

تکمیل کد:

سوالی با توضیح مسئله داده می‌شود و کاربر باید کد کامل را بنویسد.

مثال: "یک حلقه بنویسید که اعداد ۱ تا ۵ را چاپ کند."

تشخیص خطا:

قطعه کدی نمایش داده می‌شود و کاربر باید خطای آن را پیدا کند.

مثال:

```
for i in range(5)
```

```
    print(i)
```

مرتب‌سازی خطوط کد:

خطوط کد به ترتیب اشتباه نمایش داده می‌شوند و کاربر باید آن‌ها را به ترتیب درست مرتب کند.

مثال:

print(i)

for i in range(5):

پاسخ چندگزینه‌ای:

سوالی درباره مفاهیم درس مطرح می‌شود و کاربر از بین گزینه‌ها پاسخ صحیح را انتخاب می‌کند.

مثال: "حلقه for برای چه منظوری استفاده می‌شود؟"

اجرای کد و مشاهده خروجی:

کاربر باید کدی را اجرا کند و خروجی صحیح را پیش‌بینی کند یا اصلاح کند.

۳. تعامل و بازخورد:

بازخورد فوری:

بلافاصله پس از ارسال پاسخ، نتیجه نمایش داده شود:

پاسخ صحیح: پیام تشویقی (مثل "آفرین، درست بود!")

پاسخ اشتباه: توضیحی کوتاه برای اصلاح پاسخ.

سیستم امتیازدهی:

هر پاسخ صحیح به کاربر امتیاز یا جم می‌دهد.

نمایش امتیازات در بالای صفحه برای ایجاد انگیزه.

۴. بخش پایانی تمرین:

نتایج تمرین:

نمایش تعداد پاسخ‌های صحیح و اشتباه.

نمایش درصد پیشرفت در درس.

پیشنهاد بازگشت به آموزش:

اگر کاربر سوالات زیادی را اشتباه پاسخ دهد، پیشنهاد بازگشت به بخش آموزش داده می‌شود.

دکمه بازگشت به مسیر یادگیری:

دکمه‌ای برای بازگشت به مسیر یادگیری یا شروع تمرین بعدی.

طراحی بصری پیشنهادی:

بخش‌بندی واضح:

سوالات به صورت کارت‌های جداگانه نمایش داده شوند.

هر کارت دارای فضای مناسب برای پاسخ‌دهی باشد.

استفاده از رنگ‌ها:

پاسخ‌های صحیح و اشتباه با رنگ‌های مشخص (سبز و قرمز) نمایش داده شوند.

نمایش وضعیت تمرین:

در بالای صفحه، تعداد سوالات باقی‌مانده و درصد پیشرفت نمایش داده شود.

مزایای این طراحی:

تنوع تمرین‌ها: سوالات متنوع باعث می‌شود کاربران با روش‌های مختلف مفاهیم را مرور کنند.

تعامل بالا: بازخورد فوری و سیستم امتیازدهی کاربران را درگیر نگه می‌دارد.

تقویت یادگیری: ترکیب تمرین‌های کدنویسی و سوالات مفهومی به تثبیت مفاهیم کمک

می‌کند.

هدف اصلی صفحه پروفایل

نمایش اطلاعات کلی کاربر (نام، سطح، دستاوردها، و پیشرفت).

ایجاد انگیزه از طریق نمایش مدال‌ها، آمار، و دستاوردهای کاربر.

فراهم کردن امکان ویرایش اطلاعات شخصی.

ساختار پیشنهادی صفحه پروفایل

۱. بخش بالای صفحه: اطلاعات کلی کاربر

نمایش نام و عکس پروفایل:

نام و عکس کاربر در بالای صفحه به صورت برجسته نمایش داده شود.

یک دکمه برای ویرایش اطلاعات شخصی در کنار آن قرار گیرد.

سطح و امتیازات:

نمایش سطح فعلی کاربر (Level) و نوار پیشرفت به سطح بعدی.

مجموع امتیازات یا جمع‌های کسب‌شده.

۲. بخش دستاوردها و مدال‌ها

نمایش مدال‌ها:

مدال‌های کسب‌شده در یک گرید مرتب نمایش داده شوند.

مدال‌های قفل‌شده (که هنوز کسب نشده‌اند) به صورت خاکستری نمایش داده شوند.

امکان کلیک روی هر مدال برای دیدن جزئیات (مثلاً شرایط کسب مدال).

چالش‌های تکمیل‌شده:

تعداد چالش‌های روزانه، هفتگی، و ویژه که کاربر تکمیل کرده است.

۳. بخش آمار یادگیری

زمان یادگیری:

نمایش مجموع زمان صرف‌شده در برنامه (مثلاً "۵ ساعت و ۳۰ دقیقه").

تعداد سوالات پاسخ‌داده‌شده:

نمایش تعداد کل سوالات پاسخ‌داده‌شده و درصد پاسخ‌های صحیح.

پیشرفت در دوره‌ها:

نمایش درصد پیشرفت در هر دوره.

۴. تنظیمات و ویرایش اطلاعات

ویرایش اطلاعات کاربر:

امکان تغییر نام، عکس پروفایل، زبان پیش‌فرض.

تنظیمات برنامه:

تنظیم اعلان‌ها، اهداف یادگیری روزانه، و زبان رابط کاربری.

حساب کاربری:

اطلاعات مربوط به ایمیل یا شماره تلفن برای مدیریت حساب.

طراحی بصری پیشنهادی

بخش‌بندی منظم:

صفحه به سه بخش اصلی تقسیم شود: اطلاعات کلی، دستاوردها، و آمار.

استفاده از رنگ‌ها:

رنگ‌های روشن و متنوع برای مدال‌ها.

پس‌زمینه ساده برای تمرکز بر اطلاعات.

آیکون‌ها و گرافیک:

استفاده از آیکون‌های جذاب برای هر بخش (مدال‌ها، چالش‌ها، آمار).

ویژگی‌های تعاملی:

جزئیات مدال‌ها:

کاربر با کلیک روی هر مدال می‌تواند توضیحات کامل درباره نحوه کسب آن را ببیند.

دکمه ویرایش:

امکان تغییر سریع اطلاعات شخصی.

نمایش امتیازات:

جمع امتیازات کسب‌شده به همراه گراف ساده‌ای از پیشرفت کلی.

وظایف تیمی برای پیاده‌سازی صفحه پروفایل

پیشنهاد نهایی:

این صفحه به عنوان یک مرکز انگیزشی عمل کند که کاربران بتوانند پیشرفت، دستاوردها، و

اطلاعات خود را ببینند.

طراحی باید ساده، تعاملی، و جذاب باشد تا کاربران تشویق به ادامه یادگیری شوند.

صفحه داشبورد (Dashboard)

هدف صفحه داشبورد

صفحه داشبورد به عنوان مرکز مدیریت یادگیری طراحی شده است و اطلاعات کلیدی درباره پیشرفت، دوره‌های فعال، و پیشنهادات ویژه را در یک مکان به کاربران ارائه می‌دهد. این صفحه کاربران را برای مدیریت روزانه فعالیت‌های یادگیری و مشاهده پیشرفت‌های خود ترغیب می‌کند.

ساختار و جزئیات صفحه داشبورد

۱. بخش بالای صفحه: اطلاعات کلی روزانه

۱. خوشامدگویی:

- یک پیام شخصی‌سازی شده برای کاربر نمایش داده شود.
- مثال: "سلام، علی! آماده‌ای امروز یادگیری رو شروع کنی؟"

۲. آمار روزانه:

- تعداد درس‌های تکمیل شده امروز.
- تعداد سوالات پاسخ داده شده.
- مجموع امتیازات یا جمع‌های کسب شده.

۳. نوار پیشرفت کلی:

- یک نوار درصدی که نشان‌دهنده پیشرفت کاربر در مسیر یادگیری کلی است.
- کاربر با کلیک روی این نوار می‌تواند جزئیات بیشتری از دوره‌های خود را ببیند.

۲. بخش میانی: دوره‌های فعال

۱. کارت‌های دوره:

- هر دوره در یک کارت جداگانه نمایش داده شود.
- اطلاعات کلیدی هر کارت:
 - نام دوره (مثل "پایتون مقدماتی").
 - درصد پیشرفت در دوره.
 - تعداد درس‌های باقی‌مانده.
- عملکرد کارت: با کلیک روی هر کارت، کاربر به مسیر یادگیری مربوط به آن دوره هدایت می‌شود.
- ۲. اضافه کردن دوره جدید:
 - یک دکمه برجسته با آیکون "+" برای افزودن دوره‌های جدید در پایین یا کنار کارت‌ها قرار گیرد.

۳. بخش پیشنهادات ویژه

۱. درس‌های تقویتی:

- بر اساس نقاط ضعف کاربر در دوره‌ها، درس‌های تقویتی پیشنهاد داده شوند.
- مثال: "چالش تقویتی برای حلقه‌ها."

۲. دوره‌های جدید:

- پیشنهاد دوره‌های جدید مرتبط با علایق و نیازهای کاربر.

۴. دسترسی سریع به چالش‌ها

۱. کارت یا دکمه دسترسی به صفحه چالش‌ها:

- در انتهای صفحه یا در کنار پیشنهادات ویژه، دکمه‌ای با عنوان "شروع چالش‌های روزانه!" قرار داده شود.

- این دکمه کاربر را به صفحه جداگانه چالش‌ها هدایت می‌کند.

۲. نمایش تعداد چالش‌های فعال:

- اگر چالش‌ها نقش بسیار پررنگ‌تری پیدا کنند، تعداد چالش‌های فعال (مثلاً: ۲ "چالش فعال") به صورت خلاصه در این بخش نمایش داده شود تا کاربر از وضعیت چالش‌ها مطلع باشد.

ویژگی‌های تعاملی صفحه داشبورد

۱. اطلاعات زنده:

- کارت‌ها به صورت زنده اطلاعات پیشرفت و امتیازات را به‌روزرسانی کنند.

۲. شخصی‌سازی پیشنهادات:

- بر اساس عملکرد کاربر، پیشنهادات مرتبط نمایش داده شود.

۳. دسترسی سریع:

- کاربران به راحتی به دوره‌ها، چالش‌ها، و سایر بخش‌های برنامه دسترسی داشته

باشند.

مزایای این طراحی:

۱. تمرکز بر فعالیت‌های روزانه:

- کاربران می‌توانند برنامه روزانه خود را مدیریت کنند و اطلاعات کلیدی را در یک نگاه ببینند.

۲. انگیزش کاربران:

- نمایش آمار پیشرفت و پیشنهادات تقویتی باعث افزایش انگیزه می‌شود.

۳. ساختار ساده و منظم:

- طراحی واضح و منطقی باعث می‌شود کاربران بدون سردرگمی از امکانات استفاده کنند.

جمع‌بندی:

صفحه داشبورد به عنوان مرکز فعالیت‌های یادگیری کاربران طراحی شده است و تمامی اطلاعات کلیدی مثل آمار یادگیری، دوره‌ها، و پیشنهادات ویژه را در یک مکان ارائه می‌دهد. این طراحی با حذف چالش‌ها از داشبورد و انتقال آن‌ها به صفحه‌ای جداگانه، تمرکز این صفحه را بر مدیریت یادگیری و پیشرفت روزانه کاربران تقویت می‌کند. اگر چالش‌ها نقش پررنگ‌تری پیدا کنند، می‌توان با افزودن اطلاعات خلاصه (مثل تعداد چالش‌های فعال) ارتباط بین داشبورد و صفحه چالش‌ها را تقویت کرد.

هدف صفحه چالش‌ها

صفحه چالش‌ها به عنوان یکی از کلیدی‌ترین بخش‌های گیمیفیکیشن برنامه طراحی شده است تا کاربران را به تعامل بیشتر، رقابت، و تقویت مهارت‌های خود تشویق کند. این صفحه چالش‌های روزانه، هفتگی، و ویژه را ارائه می‌دهد و اطلاعات مربوط به پاداش‌ها، رفتار دیگر کاربران، و رتبه‌بندی کاربران را نمایش می‌دهد.

ساختار و جزئیات صفحه چالش‌ها

۱. بخش بالای صفحه: اطلاعات کلیدی چالش‌ها

۱. عنوان صفحه:

- عنوان جذاب مثل "چالش‌های امروز" یا "چالش‌های من" با طراحی رنگارنگ.

۲. تایمر چالش‌های فعال:

- نمایش زمان باقی‌مانده برای هر چالش (روزانه، هفتگی).
- طراحی بصری برای تاکید بر زمان محدود (مثلاً رنگ قرمز برای زمانی که به پایان نزدیک است).

۳. پاداش چالش‌ها:

- نمایش تعداد جم، مدال، یا امتیاز قابل کسب برای هر چالش.
- کارت‌های چالش با بخش جداگانه‌ای برای نمایش پاداش.

۲. لیست چالش‌ها

۱. چالش‌های روزانه:

- تمرین‌های کوتاه برای مرور مفاهیم قبلی.
- مثال: "مرور حلقه‌ها" یا "تکمیل تمرین توابع".

۲. چالش‌های هفتگی:

- تمرین‌های جامع‌تر که مفاهیم مختلفی را شامل می‌شوند.
- مثال: "ساخت یک برنامه کوچک برای مدیریت لیست کارها".

۳. چالش‌های ویژه:

- مسابقات خاص یا رقابت‌های محدود با پاداش‌های بزرگ‌تر.
- رتبه‌بندی کاربران در پایان هر چالش.

۳. رفتار کاربران روی کارت چالش‌ها

۱. تعداد شرکت‌کنندگان:

- نمایش تعداد افرادی که تاکنون در این چالش شرکت کرده‌اند.
- مثال: "۱۵۰ نفر تاکنون این چالش را انجام داده‌اند".

۲. نرخ موفقیت کاربران:

- درصد موفقیت کاربران در این چالش (براساس سوالات پاسخ داده‌شده صحیح).
- مثال: "۷۵٪ از کاربران این چالش را با موفقیت تکمیل کرده‌اند".

۳. رتبه‌بندی کاربر برتر:

- نمایش کاربر یا کاربران برتر همراه با امتیاز کسب‌شده.
 - مثال: "کاربر برتر: علی با ۹۵ امتیاز".
-

۴. کارتهای چالش

۱. اطلاعات کارت:

- نام چالش (مثل "چالش مرور توابع").
- نوع چالش (روزانه، هفتگی، ویژه).
- پاداش (جم، امتیاز، مدال).
- دکمه "شروع چالش".

۲. طراحی کارت:

- رنگ‌های متنوع برای هر نوع چالش.
 - آیکن‌های مرتبط (مثل جام برای چالش‌های ویژه، ستاره برای روزانه).
-

۵. بخش پایانی: مدیریت و پیشنهاد چالش‌ها

۱. پیشنهاد چالش‌های مرتبط:

- بر اساس عملکرد کاربر در درس‌ها یا دوره‌ها، چالش‌های تقویتی پیشنهاد شوند.
- مثال: "به نظر می‌رسد در حلقه‌ها مشکل داشتی، این چالش را امتحان کن!"

۲. دسترسی به چالش‌های قبلی:

- نمایش چالش‌های تکمیل‌شده برای مرور یا مشاهده نتایج.
-

ویژگی‌های تعاملی صفحه چالش‌ها

۱. دسترسی سریع و ساده:

- کارت‌های چالش به صورت جداگانه نمایش داده شوند.

- دکمه "شروع چالش" برای هر کارت.

۲. بازخورد فوری:

- پس از تکمیل چالش، پیام تبریک یا گزارش نتایج نمایش داده شود.

۳. پاداش‌های انگیزشی:

- نمایش جم، مدال، یا امتیازات کسب‌شده بلافاصله پس از اتمام چالش.
-

طراحی بصری پیشنهادی: (UI)

۱. رنگ‌ها:

- کارت‌های رنگارنگ برای تمایز بین انواع چالش‌ها.

- تایمر چالش‌ها با رنگ‌های متنوع برای جلب توجه.

۲. فونت‌ها:

- استفاده از فونت‌های برجسته برای نام چالش و اطلاعات پاداش.

۳. آیکون‌ها:

- آیکون‌های مرتبط با چالش‌ها (جام، ساعت، مدال).

مزایای این طراحی:

۱. تمرکز مستقل: کاربران می‌توانند به صورت جداگانه روی چالش‌ها تمرکز کنند.
۲. افزایش تعامل: نمایش رفتار کاربران و پاداش‌ها باعث ترغیب بیشتر به مشارکت می‌شود.
۳. تجربه کاربری روان: طراحی جداگانه برای چالش‌ها باعث می‌شود کاربران سردرگم نشوند.

جمع‌بندی:

صفحه چالش‌ها به کاربران اجازه می‌دهد در فعالیت‌های گیمیفیکیشن محور شرکت کنند، رتبه خود را با دیگران مقایسه کنند، و از طریق پاداش‌های متنوع انگیزه بیشتری برای یادگیری پیدا کنند. با نمایش رفتار دیگر کاربران روی کارت‌های چالش، تعامل و جذابیت صفحه بیشتر شده و کاربران تشویق به شرکت در چالش‌ها می‌شوند.

صفحه جزئیات چالش (Challenge Details Page)

هدف صفحه جزئیات چالش

صفحه جزئیات چالش به کاربران امکان می‌دهد اطلاعات دقیق چالش انتخاب‌شده، سوالات و تمرین‌های مرتبط، و پاداش‌های قابل دریافت را مشاهده و مدیریت کنند. این صفحه بر ارائه بازخورد فوری، ذخیره پیشرفت، و ایجاد انگیزه از طریق نمایش پاداش‌ها تمرکز دارد.

ساختار و جزئیات صفحه جزئیات چالش

۱. بخش بالای صفحه: اطلاعات کلی چالش

۱. عنوان و توضیحات چالش:

- نمایش عنوان چالش (مثلاً: "چالش مرور حلقه‌ها").
- توضیح کوتاه درباره هدف چالش.

۲. تایمر (در صورت محدودیت زمانی):

- نمایش زمان باقی‌مانده برای تکمیل چالش.
- طراحی تایمر زنده با رنگ‌های متغیر برای تاکید بر محدودیت زمانی.

۳. پاداش چالش:

- نمایش تعداد جم، امتیاز، یا مدال قابل کسب.
- استفاده از آیکون‌ها و تصاویر کوچک برای جذاب‌تر کردن نمایش پاداش.

۲. سوالات و تمرین‌ها

۱. لیست سوالات یا تمرین‌ها:

- سوالات به ترتیب در کارتهای جداگانه نمایش داده شوند.
- هر سوال قابل کلیک و تعاملی باشد.

۲. نوع سوالات:

- پر کردن جای خالی در کد.
- تکمیل کد یا نوشتن یک قطعه کد جدید.

- تشخیص و رفع خطا.

- مرتب‌سازی خطوط کد به ترتیب درست.

۳. مدیریت پیشرفت:

- نشان دادن تعداد سوالات پاسخ‌داده‌شده و تعداد باقی‌مانده.

- استفاده از نوار پیشرفت یا درصد تکمیل چالش.

۳. بازخورد و ذخیره پیشرفت

۱. بازخورد فوری:

- بعد از ارسال پاسخ، بازخورد فوری نمایش داده شود:

- پاسخ درست: پیام تشویقی و نمایش پاسخ صحیح.

- پاسخ اشتباه: نمایش توضیح کوتاه درباره اشتباه و راه‌حل صحیح.

۲. ذخیره پیشرفت:

- در چالش‌های طولانی، پیشرفت کاربر به صورت خودکار ذخیره شود تا در صورت خروج، بتواند از همان نقطه ادامه دهد.

۴. بخش پایانی: نتایج و پاداش

۱. نمایش نتایج چالش:

- تعداد سوالات درست و اشتباه.

- درصد موفقیت کاربر.

○ مقایسه عملکرد کاربر با میانگین کاربران دیگر.

۲. پاداش‌ها:

○ نمایش جم، امتیاز، یا مدال کسب‌شده.

○ پیام تبریک برای تکمیل موفق چالش.

۳. پیشنهاد چالش بعدی:

○ پیشنهاد چالشی مرتبط برای ادامه.

ویژگی‌های تعاملی صفحه جزئیات چالش

۱. تعامل با سوالات:

○ کاربران بتوانند پاسخ‌ها را مستقیماً در کارت سوال وارد کرده یا کد خود را اجرا کنند.

۲. نمایش پیشرفت:

○ نوار پیشرفت به‌صورت زنده به‌روزرسانی شود.

۳. دسترسی سریع:

○ دکمه‌های "بازگشت به صفحه چالش‌ها" و "شروع چالش بعدی".

طراحی بصری پیشنهادی: (UI)

۱. بخش‌بندی واضح:

○ اطلاعات کلی چالش در بالای صفحه.

○ سوالات و تمرین‌ها در میانه صفحه.

○ نتایج و پاداش در انتهای صفحه.

۲. رنگ‌ها و گرافیک:

○ تایمر با رنگ‌های زنده (مثلاً سبز، زرد، قرمز).

○ کارتهای جداگانه برای سوالات با رنگ‌های آرام و خوانا.

۳. آیکون‌ها:

○ آیکون‌های کوچک برای تایمر، پاداش، و سوالات.

مزایای این طراحی:

۱. تمرکز کاربر: کاربر می‌تواند به راحتی چالش را مدیریت و پیشرفت خود را مشاهده کند.

۲. تعامل بالا: بازخورد فوری و قابلیت تعامل با سوالات، تجربه کاربری بهتری ایجاد می‌کند.

۳. حفظ انگیزه: نمایش پاداش‌ها و پیشنهاد چالش بعدی باعث تشویق کاربران به ادامه می‌شود.

جمع‌بندی:

صفحه جزئیات چالش به کاربران امکان می‌دهد با سوالات و تمرین‌های مرتبط با چالش تعامل کنند، پیشرفت خود را مشاهده کنند، و پاداش‌های خود را در پایان دریافت کنند. این صفحه با طراحی تعاملی و بازخورد فوری، تجربه کاربری جذاب و انگیزشی ایجاد می‌کند.

هدف صفحه پروژه‌ها

صفحه پروژه‌ها به عنوان یک فضای مستقل طراحی شده است تا کاربران بتوانند مفاهیم یادگرفته‌شده را در قالب پروژه‌های عملی پیاده‌سازی کنند. این صفحه با رویکرد تعاملی، کاربران را تشویق می‌کند تا مهارت‌های برنامه‌نویسی خود را تقویت کنند و بازخورد فوری دریافت کنند.

ساختار و جزئیات صفحه پروژه‌ها

۱. بخش بالای صفحه: معرفی پروژه‌ها

۱. عنوان صفحه:

- نمایش عنوان "پروژه‌های من" یا "تمرین‌های عملی" با طراحی جذاب.

۲. توضیح کوتاه:

- توضیحی درباره اهمیت پروژه‌ها در یادگیری برنامه‌نویسی.
- مثال: "پروژه‌ها به شما کمک می‌کنند مفاهیم یادگرفته‌شده را در سناریوهای واقعی پیاده کنید."

۳. نمایش تعداد پروژه‌ها:

- تعداد پروژه‌های تکمیل‌شده، در حال انجام، و پروژه‌های جدید.
-

۲. لیست پروژه‌ها

۱. نمایش پروژه‌ها به صورت کارت:

- هر پروژه در یک کارت مجزا نمایش داده شود.
- اطلاعات اصلی پروژه روی کارت:
 - عنوان پروژه (مثل "ساخت ماشین حساب").
 - سطح سختی (مقدماتی، متوسط، پیشرفته).
 - وضعیت پروژه (تکمیل شده، در حال انجام، شروع نشده).
 - درصد پیشرفت پروژه.
 - پاداش پروژه (جم یا امتیاز).
 - دکمه "ادامه پروژه" یا "شروع پروژه".

۲. دسته‌بندی پروژه‌ها:

- پروژه‌ها بر اساس سطح سختی یا موضوع دسته‌بندی شوند:
 - مقدماتی: پروژه‌های ساده برای کاربران جدید.
 - متوسط: تمرین‌های ترکیبی برای درک مفاهیم بیشتر.
 - پیشرفته: پروژه‌هایی که مهارت‌های ترکیبی و خلاقانه نیاز دارند.

۳. تعاملات صفحه پروژه‌ها

۱. پیشرفت زنده:

- نوار پیشرفت یا درصد تکمیل پروژه به صورت زنده به روزرسانی شود.

۲. بازخورد فوری:

- پس از ارسال هر بخش پروژه، بازخورد فوری نمایش داده شود (مثلاً "این بخش به درستی پیاده سازی شد!").

۳. پاداش برای تکمیل پروژه‌ها:

- نمایش جم، امتیاز یا مدال برای پروژه‌های تکمیل شده.
-

۴. آموزش تعاملی پروژه‌ها

۱. مرحله به مرحله:

- هر پروژه به مراحل کوچک تقسیم شود.
- مثال: "مرحله ۱: تعریف متغیرها" یا "مرحله ۲: ایجاد توابع محاسبه گر".

۲. فضای کدنویسی تعاملی:

- کاربران بتوانند در هر مرحله کد مربوطه را بنویسند و اجرا کنند.
- بازخورد فوری برای اصلاح اشتباهات یا تایید پاسخ صحیح ارائه شود.

۳. ذخیره پیشرفت:

- امکان ذخیره پیشرفت در پروژه و بازگشت به آن در زمان دیگر.
-

۵. بخش پایانی صفحه: مدیریت پروژه‌ها

۱. پروژه‌های پیشنهادی:

- نمایش پروژه‌های جدید بر اساس پیشرفت کاربر.
- مثال: "پروژه‌ای برای تقویت حلقه‌ها و توابع".

۲. وضعیت پروژه‌ها:

- امکان مشاهده تاریخچه پروژه‌های تکمیل‌شده.
 - نمایش لیست پروژه‌هایی که کاربر در حال انجام آن‌هاست.
-

ویژگی‌های تعاملی پروژه‌ها

۱. مرحله‌به‌مرحله و بازخورد فوری:

- هر مرحله مستقل باشد و کاربران بتوانند بازخورد فوری دریافت کنند.

۲. فضای کدنویسی تعاملی:

- ارائه یک محیط ساده برای نوشتن و اجرای کد در هر مرحله.

۳. ذخیره پیشرفت:

- امکان ذخیره پیشرفت پروژه و بازگشت به آن در هر زمان.
-

طراحی بصری پیشنهادی: (UI)

۱. رنگ‌ها:

- رنگ‌های ملایم برای دسته‌بندی پروژه‌ها (سبز برای تکمیل‌شده، آبی برای در حال انجام).

۲. آیکون‌ها:

- آیکون‌های کوچک برای نشان دادن سطح سختی (مثلاً ستاره برای پیشرفته).

۳. فونت‌ها:

○ فونت‌های واضح و خوانا برای عناوین پروژه‌ها و اطلاعات کلیدی.

مزایای این طراحی:

۱. یادگیری تعاملی: کاربران مفاهیم را حین انجام پروژه یاد می‌گیرند.
 ۲. بازخورد فوری: کاربران می‌توانند در هر مرحله خطاهای خود را اصلاح کنند.
 ۳. حس موفقیت: نمایش پیشرفت و پاداش‌ها انگیزه بیشتری ایجاد می‌کند.
-

جمع‌بندی:

صفحه پروژه‌ها به کاربران اجازه می‌دهد تا مفاهیم برنامه‌نویسی را در قالب پروژه‌های مستقل و عملی پیاده‌سازی کنند. این صفحه با طراحی تعاملی و نمایش پیشرفت کاربران، تجربه‌ای حرفه‌ای و انگیزشی ارائه می‌دهد.

صفحه جزئیات پروژه (Project Details Page)

هدف صفحه جزئیات پروژه

صفحه جزئیات پروژه به کاربران امکان می‌دهد تا پروژه انتخابی خود را به صورت تعاملی و مرحله‌به‌مرحله انجام دهند. این صفحه نه تنها آموزش ساخت پروژه را ارائه می‌دهد، بلکه به

کاربران فرصت می‌دهد کدنویسی کنند، بازخورد فوری بگیرند و مهارت‌های خود را تقویت کنند.

ساختار و جزئیات صفحه جزئیات پروژه

۱. بخش بالای صفحه: اطلاعات کلی پروژه

۱. عنوان و توضیحات پروژه:

- نمایش عنوان پروژه (مثلاً: "ساخت ماشین حساب").
- توضیح کوتاه درباره هدف پروژه و مفاهیمی که پوشش داده می‌شود.
- مثال: "این پروژه به شما کمک می‌کند تا مفاهیم توابع و شرطها را در عمل به کار بگیرید."

۲. نوار پیشرفت:

- نمایش درصد تکمیل پروژه به صورت بصری.
- هر مرحله کامل شود، پیشرفت نوار به‌روزرسانی شود.

۳. پاداش پروژه:

- نمایش جم، امتیاز، یا مدال قابل دریافت در پایان پروژه.
-

۲. بخش میانی صفحه: مراحل پروژه (تعامل مرحله به مرحله)

۱. نمایش مراحل پروژه:

- پروژه به چند مرحله کوچک تقسیم شود.
- هر مرحله دارای:

- توضیح کوتاه (هدف مرحله).
- نمونه کد یا استارتر کد. (Starter Code)
- دستورالعمل واضح برای کاری که باید انجام شود.

۲. فضای تعاملی کدنویسی:

- کاربر در هر مرحله بتواند کد خود را بنویسد یا تکمیل کند.
- امکان اجرای کد و مشاهده خروجی.

۳. بازخورد فوری:

- پس از اجرای کد:
- اگر درست باشد: پیام تبریک و نمایش مرحله بعد.
- اگر اشتباه باشد: پیام راهنما و توضیح درباره مشکل.

۴. ذخیره پیشرفت:

- در هر مرحله، پیشرفت کاربر به صورت خودکار ذخیره شود.
- کاربران بتوانند از همان مرحله ادامه دهند.

۳. بخش پایانی: تکمیل پروژه

۱. نتیجه پروژه:

- نمایش کد کامل پروژه پس از تکمیل.
- امکان اجرای پروژه کامل و مشاهده خروجی نهایی.

۲. ارزیابی پروژه:

- نمایش تعداد مراحل درست تکمیل شده.

- ارائه پاداش نهایی (جم، امتیاز).

۳. پیشنهاد پروژه‌های مرتبط:

- نمایش پروژه‌هایی که کاربر می‌تواند پس از این پروژه انجام دهد.

ویژگی‌های تعاملی صفحه جزئیات پروژه

۱. محیط کدنویسی تعاملی:

- فضای ساده و واضح برای نوشتن، ویرایش و اجرای کد.

- پشتیبانی از ابزارهایی مانند راهنما (Hints) یا تکمیل خودکار.

۲. بازخورد فوری:

- پس از هر اجرای کد، بازخورد فوری درباره درستی یا اشتباه بودن کد.

- ارائه پیشنهادات برای اصلاح اشتباهات.

۳. نمایش پیشرفت:

- نوار پیشرفت زنده و نشانگر تعداد مراحل تکمیل شده.

طراحی بصری پیشنهادی: (UI)

۱. بخش‌بندی مراحل:

- هر مرحله در یک کارت یا باکس مجزا نمایش داده شود.

۲. نوار پیشرفت:

○ درصد تکمیل پروژه به صورت بصری در بالای صفحه.

۳. رنگ‌ها و آیکون‌ها:

○ رنگ‌های آرام و شاد برای مراحل مختلف.

○ آیکون‌هایی برای تایید (تیک سبز) یا خطا (ضربدر قرمز).

مزایای این طراحی:

۱. یادگیری تعاملی: کاربران می‌توانند مفاهیم را حین انجام پروژه یاد بگیرند.

۲. بازخورد فوری: کاربران می‌توانند در هر مرحله خطاهای خود را اصلاح کنند.

۳. حس موفقیت: نمایش پیشرفت و پاداش‌ها انگیزه بیشتری ایجاد می‌کند.

جمع‌بندی:

صفحه جزئیات پروژه، فضایی تعاملی برای آموزش مرحله‌به‌مرحله ساخت پروژه‌ها فراهم می‌کند. کاربران نه تنها مفاهیم را یاد می‌گیرند، بلکه می‌توانند با اجرای کد خود، تجربه عملی کسب کنند و حس موفقیت بیشتری را تجربه کنند.

صفحه تنظیمات و اهداف یادگیری (Settings & Goals Page)

هدف صفحه تنظیمات و اهداف یادگیری

این صفحه به کاربران امکان می‌دهد برنامه را شخصی‌سازی کنند، اهداف روزانه خود را تنظیم کنند، و تنظیمات مربوط به اعلان‌ها و حساب کاربری را مدیریت نمایند. طراحی این صفحه باید ساده و کاربردی باشد تا تجربه کاربری بهتری ارائه دهد.

ساختار و جزئیات صفحه

۱. بخش اهداف یادگیری (Learning Goals)

۱. تنظیم زمان یادگیری روزانه:

- نمایش گزینه‌هایی برای انتخاب مدت زمان یادگیری:

▪ مثال: ۱۰، ۲۰، ۳۰ دقیقه یا گزینه دلخواه.

- نمایش نوار پیشرفت یا نمودار کوچک برای تحقق هدف روزانه.

۲. یادآوری اهداف:

- امکان فعال/غیرفعال کردن یادآوری اهداف.

- انتخاب زمان ارسال یادآوری‌ها (مثلاً: "ساعت ۸ شب").
-

۲. بخش مدیریت اعلان‌ها (Notifications)

۱. یادآوری یادگیری:

- پیام‌هایی برای یادآوری اهداف روزانه یادگیری.

▪ مثال: "فراموش نکن امروز ۱۰ دقیقه کدنویسی کنی!"

۲. اعلان‌های مربوط به چالش‌ها و مسابقات:

- فعال/غیرفعال کردن اعلان‌های مربوط به چالش‌های روزانه و هفتگی.

○ اطلاع‌رسانی نزدیک شدن به پایان زمان چالش‌ها.

▪ مثال: "چالش امروز فقط ۲ ساعت دیگر فعال است!"

۳. اعلان‌های پیشرفت و پاداش:

○ نمایش دستاوردهای کاربر و اطلاع‌رسانی درباره پاداش‌ها.

▪ مثال: "آفرین! امروز هدف‌ت رو تکمیل کردی و ۵ جم به دست آوردی".

۴. اعلان‌های پروژه‌ها:

○ اعلان‌های مرتبط با پروژه‌های در حال انجام یا پیشنهاد پروژه‌های جدید.

▪ مثال: "پروژه ساخت ماشین حساب منتظرته! ادامه بده".

۵. پیشنهادات ویژه:

○ پیام‌هایی برای تشویق کاربر به شرکت در فعالیت‌های جدید.

▪ مثال: "یک چالش ویژه برای تو داریم! ۱۰ جم ببر".

۶. انتخاب نوع اعلان:

○ انتخاب بین اعلان‌های صوتی، بصری یا هر دو.

۷. زمان‌بندی هوشمند:

○ ارسال نوتیفیکیشن در زمان‌های مناسب، مثلاً عصر یا زمانی که کاربر معمولاً از

برنامه استفاده می‌کند.

۳. بخش تنظیمات شخصی‌سازی (Personalization)

۱. ویرایش اطلاعات حساب:

- تغییر نام کاربری، ایمیل، شماره موبایل.

- امکان تغییر یا آپلود عکس پروفایل.

۲. تغییر زبان برنامه:

- انتخاب زبان رابط کاربری.

- نمایش زبان‌های موجود (مثلاً فارسی، انگلیسی).

۳. تم برنامه:

- امکان انتخاب تم‌های مختلف (مثل روشن، تاریک).

۴. بخش مدیریت حساب کاربری (Account Management)

۱. بازیابی اطلاعات:

- امکان تغییر رمز عبور.

۲. مدیریت حساب:

- مشاهده وضعیت اشتراک (در صورت وجود).

- گزینه حذف یا غیرفعال کردن حساب کاربری.

ویژگی‌های تعاملی صفحه

۱. ذخیره تغییرات فوری:

- تنظیمات پس از تغییر بلافاصله ذخیره شوند.

۲. پیغام تایید:

○ نمایش پیام تایید بعد از اعمال تغییرات (مثلاً: "تغییرات ذخیره شد!")

۳. بازگشت به تنظیمات پیش فرض:

○ امکان بازنشانی تمام تنظیمات به حالت اولیه.

طراحی بصری پیشنهادی: (UI)

۱. بخش بندی واضح:

○ هر بخش در کارت یا تب جداگانه نمایش داده شود.

۲. استفاده از سوییچ ها و دکمه ها:

○ فعال/غیرفعال کردن گزینه ها با سوییچ.

○ دکمه های ساده برای ذخیره تغییرات.

۳. رنگ ها:

○ رنگ های آرام برای پس زمینه.

○ رنگ های مشخص برای سوییچ ها و دکمه های فعال.

مزایای این طراحی:

۱. شخصی سازی بالا : کاربران می توانند برنامه را بر اساس نیازهای خود تنظیم کنند.

۲. مدیریت آسان اهداف : امکان مشاهده و پیگیری اهداف روزانه به راحتی.

۳. ساده و کاربردی : طراحی ساده برای استفاده راحت کاربران.

جمع‌بندی:

صفحه تنظیمات و اهداف یادگیری به کاربران امکان می‌دهد برنامه را شخصی‌سازی کنند، اهداف یادگیری خود را تنظیم کنند و از تجربه‌ای بهتر و کارآمدتر بهره ببرند. این صفحه طراحی شده تا ساده، کاربرپسند و انگیزشی باشد.