مستندسازی پروژه پایانی

درس برنامه‌نویسی وب

پروژه هاویرکشت

نام دانشجو: زهراساری

استاد درس: دکتر آرمین رشنو

نیمسال اول تحصیلی ۱۴۰۴

بهمن ۱۴۰۴

مرحله ۱ : بررسی و مستندسازیAPI ها

هدف :آشنایی با سرویس‌های سمت سرور و نحوه کار با ابزار swoagger

توضیحات :در این مرحله، ابتدا وارد محیط swoagger شدم که یک ابزار تحت وب برای تست و مستندسازیAPI هاست.

آدرس محیط سوگر پروژه به شکل زیر بود https://api-edu.havirkesht.ir/docs :

پس از ورود به این آدرس، لیست کاملی از تمامAPI های موجود در پروژه را مشاهده کردم. هر API شامل اطلاعاتی مانند نوع درخواست، آدرس، پارامترهای ورودی و خروجی بود .کارهایی که انجام دادم:

۱ .ورود به محیط سوگر: با مرورگر وارد آدرس بالا شدم و با رابط کاربری سوگر آشنا شدم .

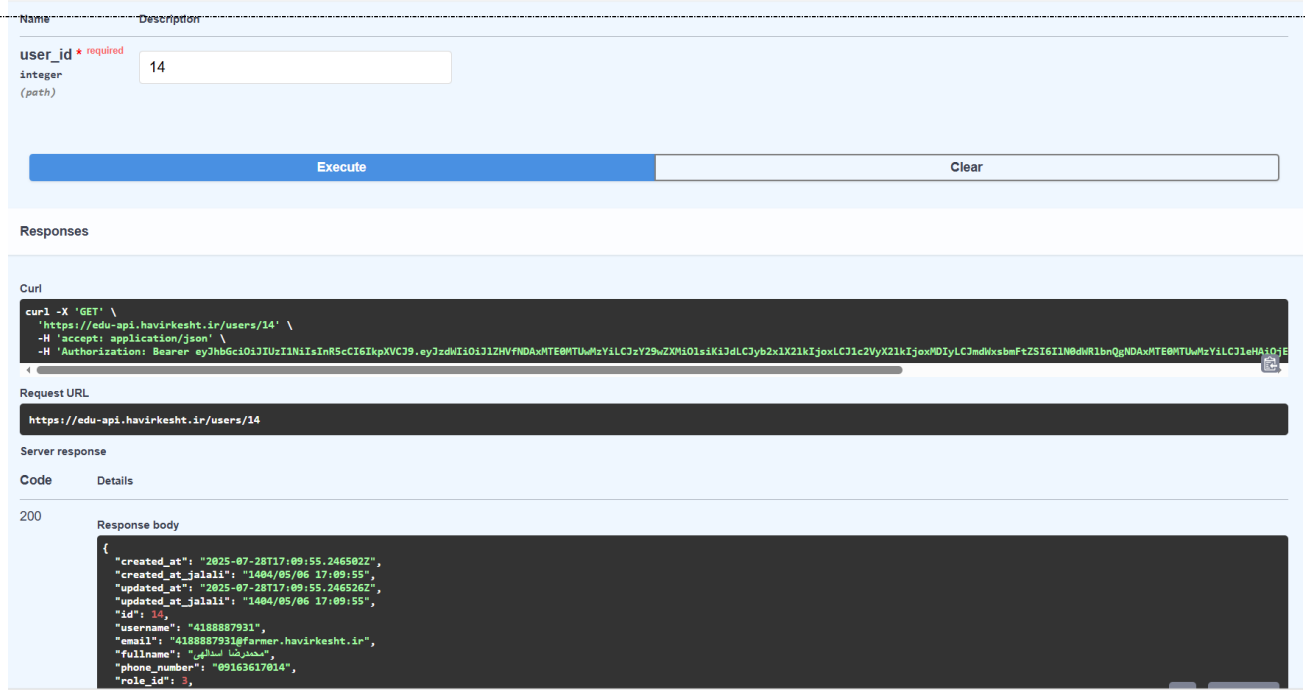
۲ .تست هر :API روی هر API کلیک کردم و دکمه "Try it out" را زدم تا ببینم چه ورودی‌هایی می‌خواهد و چه خروجی‌هایی می‌دهد.

۳ .مستندسازی در جدول: برای هر API یک ردیف در جدول اکسل ایجاد کردم و اطلاعات زیر را یادداشت کردم :نام API آدرس دقیق نوع درخواست(GET ، POST، PUT، DELETE)پارامترهای ورودی نوع خروجی کاربرد و توضیحات

درفایل :API_HavirKesht.xlsx مستند سازی کردم

نام API	متد	مسیر	توضیحات
ورود کاربر	POST	/login	احراز هویت و دریافت توکن
ثبت کاربر ادمین	POST	/users/admin/	ایجاد کاربر جدید توسط ادمین
دریافت لیست کاربران	GET	/users/	دریافت تمام کاربران
دریافت اطلاعات کاربر	GET	/users/{user_id}	دریافت جزئیات یک کاربر
ویرایش کاربر	PUT	/users/{user_id}	به‌روزرسانی اطلاعات کاربر

در پایان این مرحله، جدول کاملی از تمامAPI های موجود داشتم که در مراحل بعدی برای اتصال صفحات به سرور استفاده کردم



مرحله ۲ : شناسایی ارتباط صفحات باAPI ها

هدف :یادگیری نحوه استفاده از ابزار بازرسی مرورگر برای دیباگ و بررسی ارتباط صفحات با سرور
توضیحات :در این مرحله، هر صفحه از سایت آموزشی را باز کردم و با استفاده از ابزار Developer Toolsمرورگر) که با فشردن F12 باز می‌شود(، بررسی کردم که هر صفحه با چهAPI هایی ارتباط دارد .کارهایی که انجام دادم:

۱ .ورود به سایت: وارد آدرس edu.havirkesht.irشدم و با نام کاربری و رمز عبور زیر وارد شدم :نام کاربری user_edu1 :رمز عبور pass_edu123 :

۲ .باز کردن ابزار بازرسی: کلید F12 را فشار دادم تا پنجره Developer Tools باز شود

۳ .رفتن به تب :Network در پنجره باز شده، روی تب "Network" کلیک کردم. این قسمت تمام درخواست‌های شبکه‌ای را نشان می‌دهد.

۴ .بارگذاری مجدد صفحه: با فشردن Ctrl+R صفحه را دوباره بارگذاری کردم تا تمام درخواست‌ها ثبت شوند

۵ .فیلتر کردن درخواست‌های :API روی فیلتر "XHR" یا "Fetch" کلیک کردم تا فقط درخواست‌های مربوط به API نمایش داده شوند

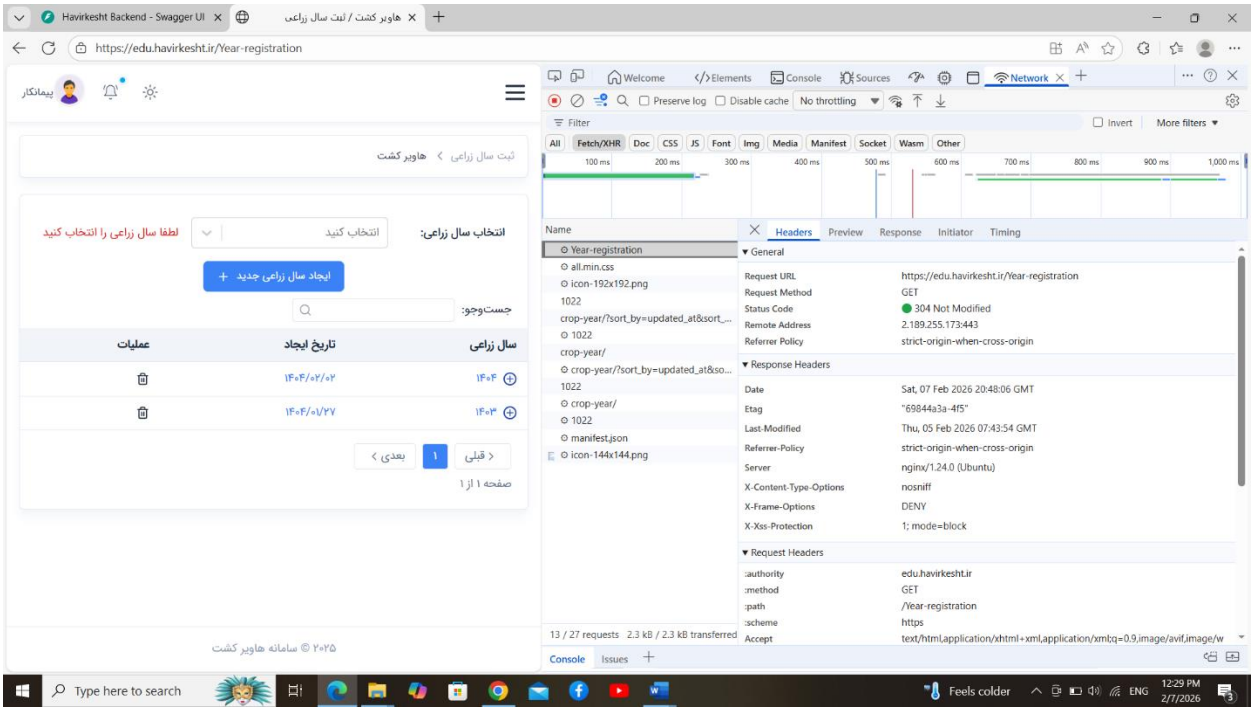
۶ .بررسی هر درخواست: روی هر درخواست کلیک کردم و موارد زیر را بررسی کردم :آدرس دقیق APIنوع درخواست داده‌های ارسالی (Request Payload) پاسخ دریافتی (Response) کدهای وضعیت (Status Code) ۷ .ثبت ارتباطات: برای هر صفحه، یادداشت کردم که کدامAPI ها فراخوانی می‌شوند

.درفایل:api_list.xlsx مستندسازی کردم

نمونه‌ای از ارتباطات شناسایی شده

نام صفحه	های مورد استفادهAPI	عملکرد
login.html	POST /login	ورود کاربر و دریافت توکن
dashboard.html	GET /users/me, GET /stats	نمایش داشبورد و آمار کاربر
provinces.html	GET /provinces/, POST /provinces/	مدیریت استان‌ها
villages.html	GET /villages/, POST /villages/	مدیریت روستاها
operations.html	GET /operations/, POST /operations/	مدیریت عملیات

تیجه :حالا می‌دانستم که هر صفحه دقیقاً با چهAPI هایی کار می‌کند و چه داده‌هایی رد و بدل می‌شود. این اطلاعات برای مرحله بعد که باید خودم صفحات را پیاده‌سازی کنم، خیلی مهم است



مرحله ۳: پیاده‌سازی صفحات در محیط محلی

هدف :ساخت صفحات سمت کاربر و اتصال آن‌ها بهAPI های سمت سرور

توضیحات :در این مرحله، تمام صفحات وب را از ابتدا پیاده‌سازی کردم. از آنجایی که پروژه اصلی با ری‌اکت نوشته نشده بود، من تصمیم گرفتم ازHTML ، CSSو جاوااسکریپت ساده استفاده کنم .ابزارها و فناوری‌های استفاده شده :HTML: برای ساختار صفحات

CSSبرای طراحی و زیباسازی

JavaScriptبرای پویایی و ارتباط با سرور

tailwind cssبرای طراحی سریع‌تر و واکنش‌گرا

Axiosبرای ارسال درخواست‌های HTTP به سرور

LocalStorageبرای ذخیره توکن ورود کاربر

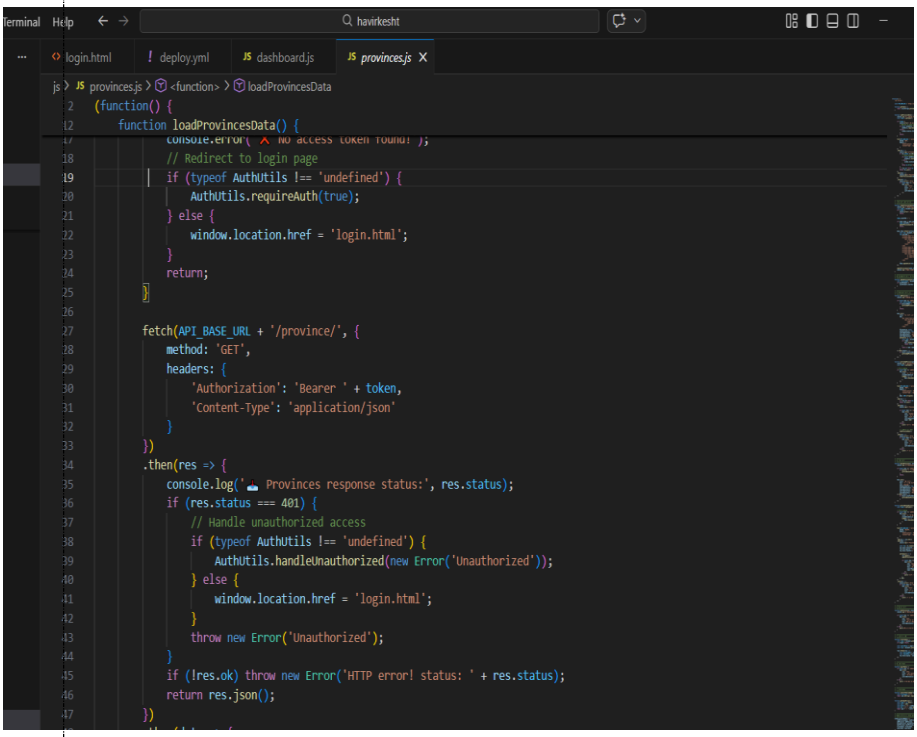
کارهایی که انجام دادم:

۱ .ساخت صفحه ورود :یک فرم ساده با دو فیلد (نام کاربری و رمز عبور) ساختم با جاوااسکریپت، هنگام کلیک روی دکمه ورود، اطلاعات را به API ورود ارسال کردم در صورت موفقیت، توکن دریافتی را در LocalStorage ذخیره کردم

۲ .ساخت صفحه اصلی (داشبورد) :اطلاعات آماری را از API دریافت و نمایش دادم جداول و نمودارها را با داده‌های دریافتی پر کردم

۳ .ساخت صفحات دیگر :هر صفحه را مطابق با طراحی اصلی پیاده‌سازی کردم هر صفحه را به APIهای مربوطه متصل کردم.

نمونه کد اتصال به API



نتیجه :در پایان این مرحله، تمام صفحات به صورت کامل پیاده‌سازی شده بودند و روی localhost اجرا می‌شدند.

کاربر می‌توانست وارد شود، اطلاعات را مشاهده کند و با سیستم کار کند.

مرحله ۴: مدیریت خطاها

هدف :نمایش پیام‌های مناسب به کاربر هنگام بروز خطا

توضیحات :وقتی کاربر با سیستم کار می‌کند، ممکن است خطاهایی رخ دهد. مثلاً اینترنت قطع شود، رمز عبور اشتباه باشد، یا سرور مشکل داشته باشد. در این مرحله، برای تمام این حالات پیام‌های مناسب پیاده‌سازی کردم

.کتابخانه استفاده شده :از کتابخانه SweetAlert2 استفاده کردم که پیام‌های زیبا و کاربرپسند نمایش می‌دهد .

انواع خطاها و پیام‌های مربوطه

۱: .خطای ورود :اگر نام کاربری یا رمز عبور اشتباه بود پیام: "نام کاربری یا رمز عبور اشتباه است

۲" .خطای شبکه :اگر اینترنت قطع بود یا سرور در دسترس نبود پیام: "خطا در برقراری ارتباط با سرور. لطفاً اتصال اینترنت خود را بررسی کنید"

۳ .خطای دسترسی :اگر کاربر بدون ورود به سیستم می‌خواست صفحات را ببیند پیام: "لطفاً ابتدا وارد سیستم شوید"

۴ .خطای اعتبارسنجی :اگر کاربر فیلدهای فرم را خالی می‌گذاشت پیام: "لطفاً تمام فیلدها را پر کنید"

مرحله ۵: تهیه دامنه وانتقال مدیریت دامنه به کلودفلر

هدف :تهیه یک دامنه اختصاصی و مدیریت آن از طریق کلودفلر

توضیحات :برای اینکه سایت من یک آدرس حرفه‌ای داشته باشد، باید یک دامنه می‌خریدم. دامنه مثل آدرس خانه است که مردم با آن شما را پیدا می‌کنند

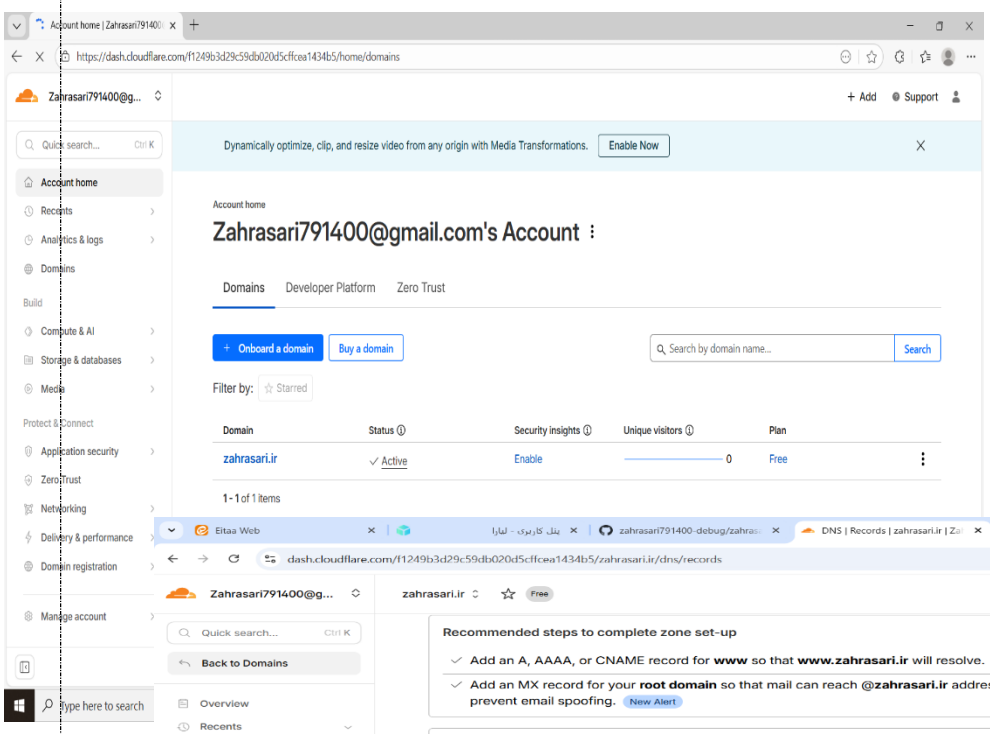
.کارهایی که انجام دادم

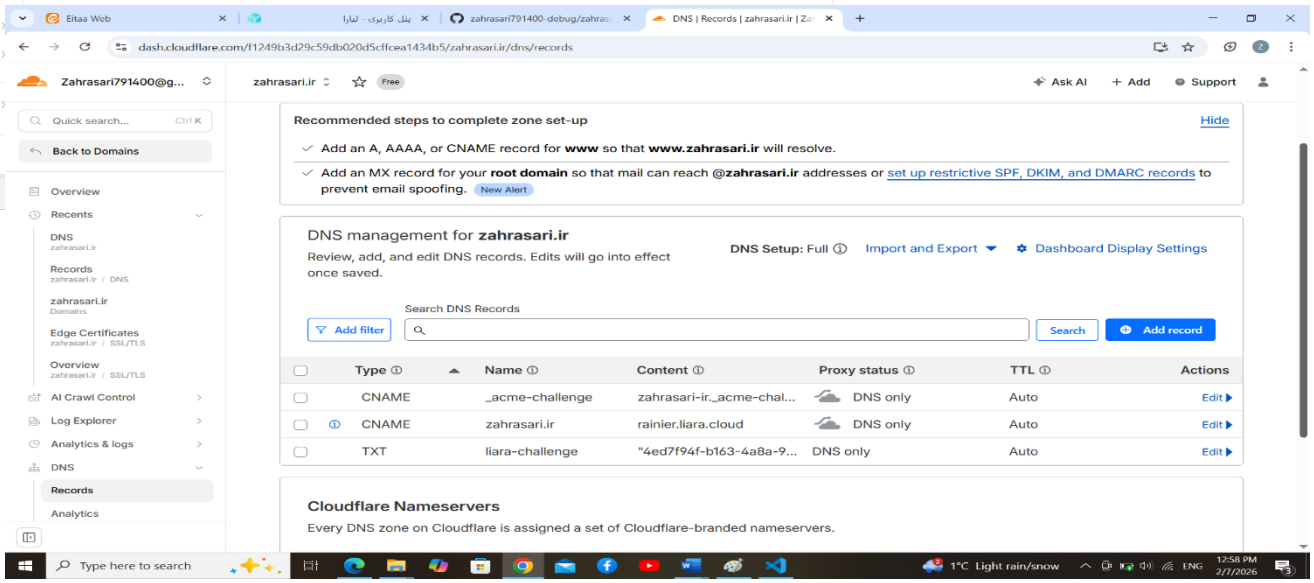
۱: .خرید دامنه :به سایت nic.irرفتم دامنه zahrasari.irرا جستجو کردم بعد از اطمینان از موجود بودن، آن را خریداری کردم

۲ .ثبت‌نام در کلودفلر :به سایت cloudflare.comرفتم یک حساب کاربری رایگان ساختم دامنه خریداری شده را به کلودفلر اضافه کردم

۳ .تغییر نیم‌سرورها :کلودفلر دو آدرس نیم‌سرور به من داد به پنل مدیریت دامنه در nic.irبرگشتم نیم‌سرورهای پیش‌فرض را با نیم‌سرورهای کلودفلر جایگزین کردم این کار ممکن است تا ۲۴ ساعت طول بکشد تا اعمال شود

۴ .تنظیمات DNS در کلودفلر :یک رکورد A با مشخصات زیر اضافه کردم Type: A : Name: @ Content: آی‌پی سرور لیارا Proxy status: فعال (نارنجی) مزایای استفاده از کلودفلر :امنیت بیشتر: کلودفلر از سایت در برابر حملات محافظت می‌کند سرعت بالاتر: فایل‌های استاتیک را کش می‌کند و سرعت بارگذاری را افزایش می‌دهد گواهی SSLرایگان: به صورت خودکار گواهی SSL صادر می‌کند آمار و گزارش: آمار بازدیدکنندگان را نشان می‌دهد





مرحله ۶: راه‌اندازی سرور

هدف :اجاره یک سرور و نصب و راه‌اندازی پروژه روی آن

توضیحات :برای اینکه سایت من در اینترنت در دسترس باشد، باید روی یک سرور قرار می‌گرفت. من از سرویس لیارا استفاده کردم

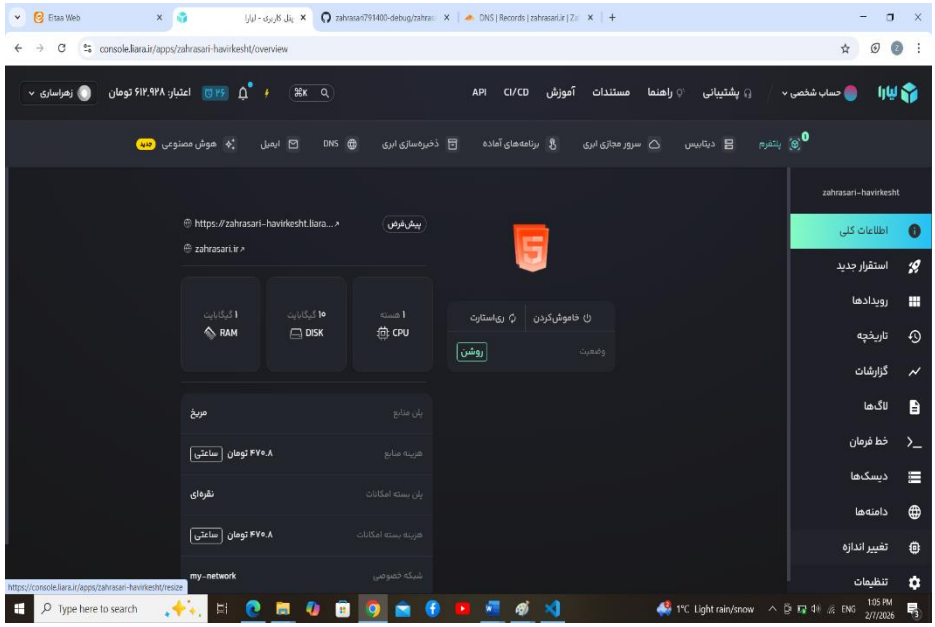
مراحل کار:

۱ .ثبت‌نام در لیارا :به سایت liara.ir رفتم با ایمیل خودم ثبت‌نام کردم حساب کاربری را تأیید کردم

۲ .ساخت اپلیکیشن جدید :روی دکمه "ایجاد برنامه" کلیک کردم نوع پلتفرم را "Static" انتخاب کردم) چون فقط HTML/CSS/JS داشتم (نام برنامه را zahrasari-havirkesht انتخاب کردم منطقه را "ایران" انتخاب کردم

۳ .آپلود فایل‌ها :در ابتدا فایل‌ها را به صورت دستی آپلود کردم تا مطمئن شوم همه چیز درست کار می‌کند :تمام فایل‌های پروژه را فشرده کردم (zip) در پنل لیارا آپلود کردم منتظر ماندم تا عملیات deploy تمام شود

۴ .تنظیم دامنه :در بخش "دامنه‌ها" روی "افزودن دامنه" کلیک کردم دامنه zahrasari.ir را اضافه کردم لیارا یک آدرس CNAME به من داد به کلودفلر برگشتم و یک رکورد CNAMEاضافه کردم ۵ .تست :به آدرس zahrasari.ir رفتم سایت به درستی بارگذاری شد تمام قسمت‌ها را تست کردم



مرحله ۷: استفاده از گیت‌هاب

هدف :آشنایی با سیستم کنترل نسخه و استفاده از گیت‌هاب

توضیحات :گیت‌هاب یک سایت است که کدهای برنامه‌نویسی را ذخیره می‌کند و به ما کمک می‌کند تغییرات را مدیریت کنیم. مثل یک فضای ابری برای کدهاست

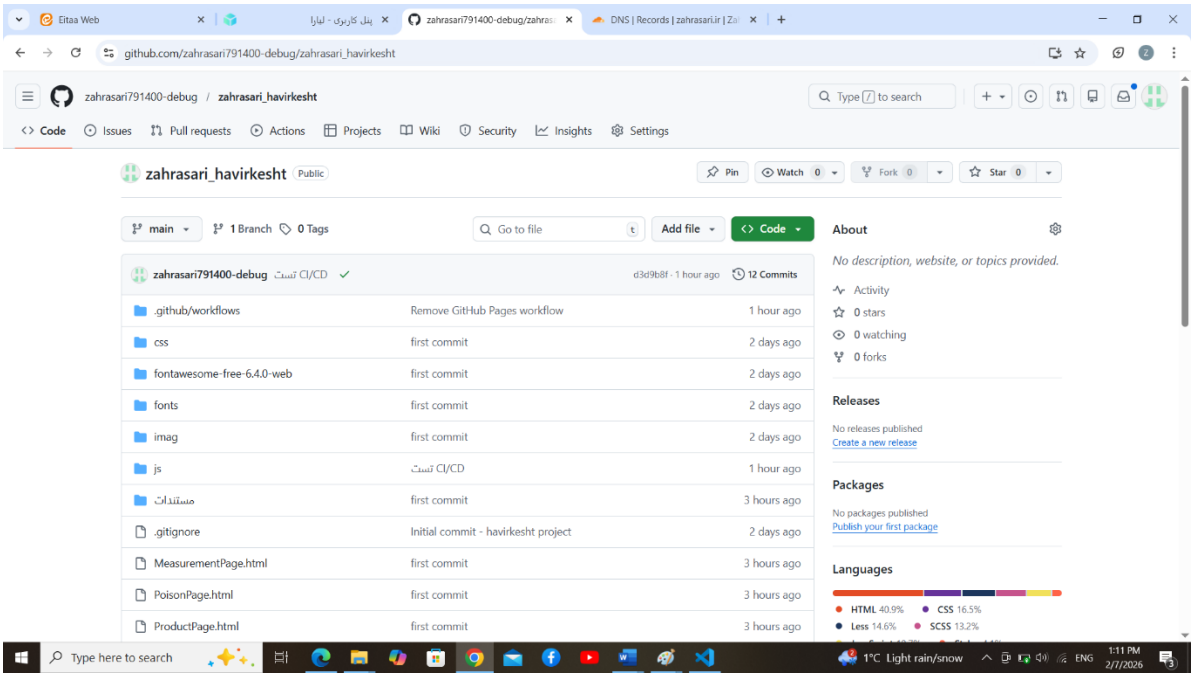
.نصب و راه‌اندازی :۱ .نصب گیت :به سایت git-scm.comرفتم نسخه ویندوز را دانلود و نصب کردم برای اطمینان، در Command Prompt دستور زیر را زدم-- git :

version۲ .تنظیم اطلاعات کاربری git config --global user.name "Zahra :

"Sari۳" git config --global user.email "zahrasari@example.com .ساخت حساب کاربری گیت‌هاب :به سایت github.comرفتم با ایمیل خودم ثبت‌نام کردم

۴ .ساخت مخزن :(Repository) روی دکمه "New repository" کلیک کردم نام مخزن را zahrasari_havirkesht گذاشتم گزینه Public را انتخاب کردم روی "Create repository"کلیک کردم ارسال کدها به گیت‌هاب :۱ .راه‌اندازی گیت در پروژه :در پوشه پروژه، Command Promptرا باز کردم و دستورات زیر را زدم : git init git add . git commit -m "اولین ارسال - پروژه کامل" ۲. اتصال به گیت‌هاب- https://github.com/zahrasari791400 git remote add origin : debug/zahrasari_havirkesht.git git branch -M main git push -u origin

main۳ .ارسال تغییرات بعدی :هر بار که تغییری می‌دادم، این دستورات را می‌زدم git : add . git commit -m "توضیح تغییرات git push origin main "مزایای استفاده از گیت‌هاب :پشتیبان‌گیری خودکار: کدها در فضای ابری ذخیره می‌شوند تاریخچه تغییرات: می‌توانم ببینم چه زمانی چه تغییری داده‌ام بازگشت به نسخه قبل: اگر اشتباهی کردم، می‌توانم به نسخه قبل برگردم همکاری تیمی: چند نفر می‌توانند روی یک پروژه کار کنند CI/CD:امکان خودکارسازی استقرار پروژه



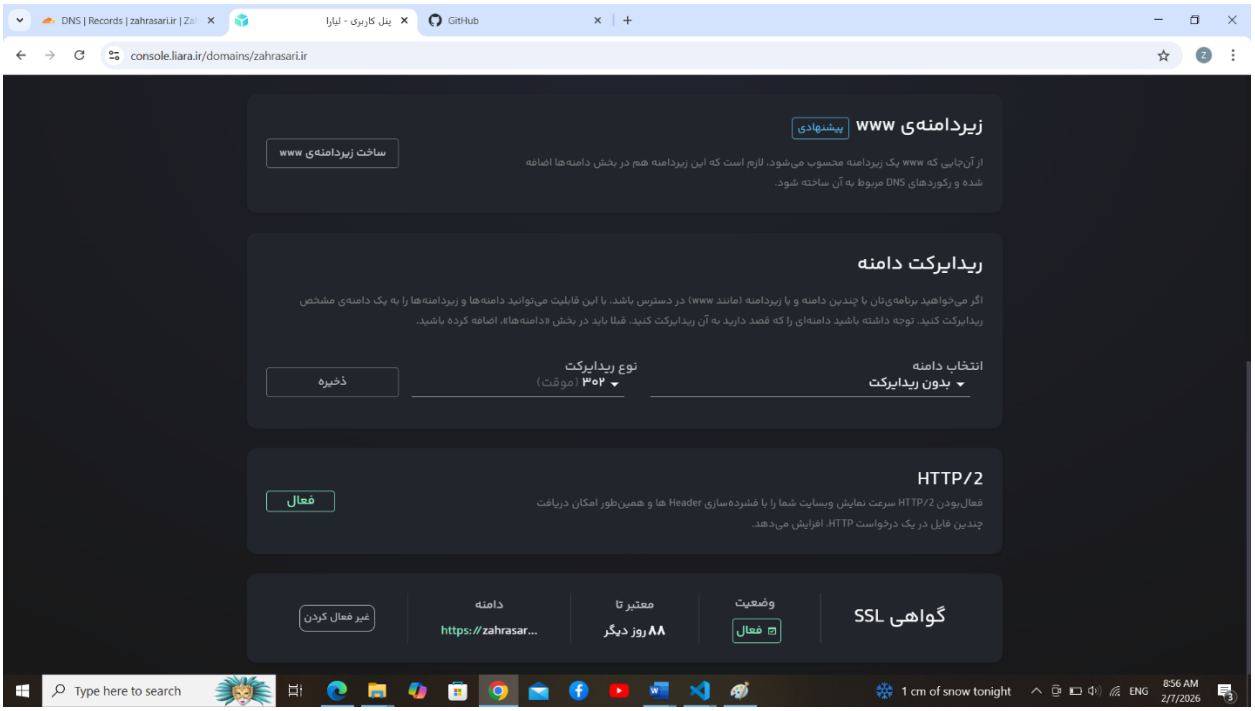
مرحله ۸: نصب گواهی امنیتی

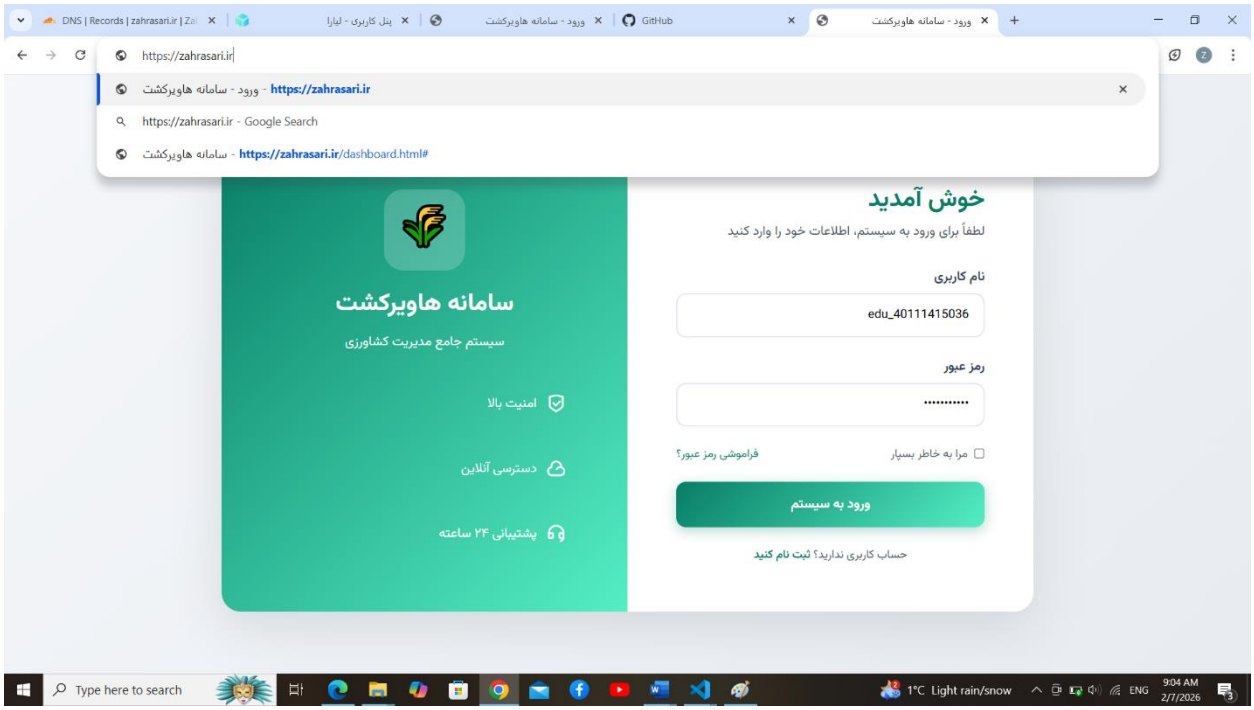
هدف :فعال کردن پروتکل HTTPS برای امنیت بیشتر

توضیحات :گواهی SSL باعث می‌شود که ارتباط بین مرورگر کاربر و سرور رمزنگاری شود. وقتی سایت گواهی SSL دارد، آدرس آن با https شروع می‌شود و یک قفل کنار آدرس نمایش داده می‌شود .چرا SSL مهم است؟ امنیت: اطلاعات کاربران (مثل رمز عبور) رمزنگاری می‌شوند اعتماد: کاربران به سایت‌های HTTPS بیشتر اعتماد دارند سئو: گوگل سایت‌های HTTPS را بهتر رتبه‌بندی می‌کند الزامی: بدونSSL ، مرورگرها سایت را "ناامن" نشان می‌دهند نحوه فعال‌سازی :خوشبختانه در لیارا و کلودفلر، این کار خیلی ساده است :۱ .در کلودفلر :به تنظیمات SSL/TLS رفتم حالت را روی "Full" گذاشتم گواهی SSLبه صورت خودکار فعال شد

۲ .در لیارا :لیارا به صورت پیش‌فرض گواهی SSL رایگان از Let's Encrypt صادر می‌کند کاری نیازی نبود، همه چیز خودکار بود ۳ .تست :به آدرس https://zahrasari.irرفتم قفل سبز کنار آدرس نمایش داده شد روی قفل کلیک کردم و جزئیات گواهی را مشاهده کردم اجبار استفاده از :HTTPS برای اینکه کاربران حتماً از

HTTPSاستفاده کنند، در کلودفلر این تنظیم را فعال کردم :به قسمت SSL/TLS → Edge Certificatesرفتم گزینه "Always Use HTTPS" را فعال کردم حالا اگر کسی با httpبه سایت می‌آمد، به صورت خودکار به https هدایت می‌شد





مرحله ۹: پیاده‌سازی CI/CD

هدف :خودکارسازی فرآیند استقرار پروژه پس از هر تغییر توضیحات :قبلاً هر بار که می‌خواستم تغییری در سایت بدهم، باید این کارها را می‌کردم:

۱ .کدها را تغییر می‌دادم

۲ .فایل‌ها را فشرده می‌کردم

۳ .وارد پنل لیارا می‌شدم

۴ .فایل زیپ را آپلود می‌کردم

۵ .منتظر می‌ماندم تا deploy شود

این کار خیلی وقت‌گیر بود و احتمال اشتباه هم بالا بود. با CI/CD این فرآیند کاملاً خودکار می‌شود.

قدم دوم: ذخیره توکن در گیت‌هاب

۱ .وارد مخزن گیت‌هاب خودم شدم

۲ .روی تب "Settings" کلیک کردم

۳ .از منوی سمت چپ، گزینه "Secrets and variables" را باز کردم

۴ .روی "Actions" کلیک کردم

۵ .دکمه "New repository secret" را زدم

۶ .در قسمت Name نوشتم LIARA_API_TOKEN :۷ .در قسمت Secret توکنی که از لیارا گرفته بودم را paste کردم

۸ .روی "Add secret" کلیک کردم

مراحل پیاده‌سازی :قدم اول: دریافت توکن از لیارا ۱ .وارد پنل لیارا شدم (console.liara.ir)۲ .روی آیکون کاربری در گوشه بالا سمت راست کلیک کردم ۳ . گزینه "تنظیمات حساب کاربری" را انتخاب کردم ۴ .تب "توکن‌های API" را باز کردم ۵ . روی "ساخت توکن جدید" کلیک کردم ۶ .نام توکن را "github-actions" گذاشتم ۷ .

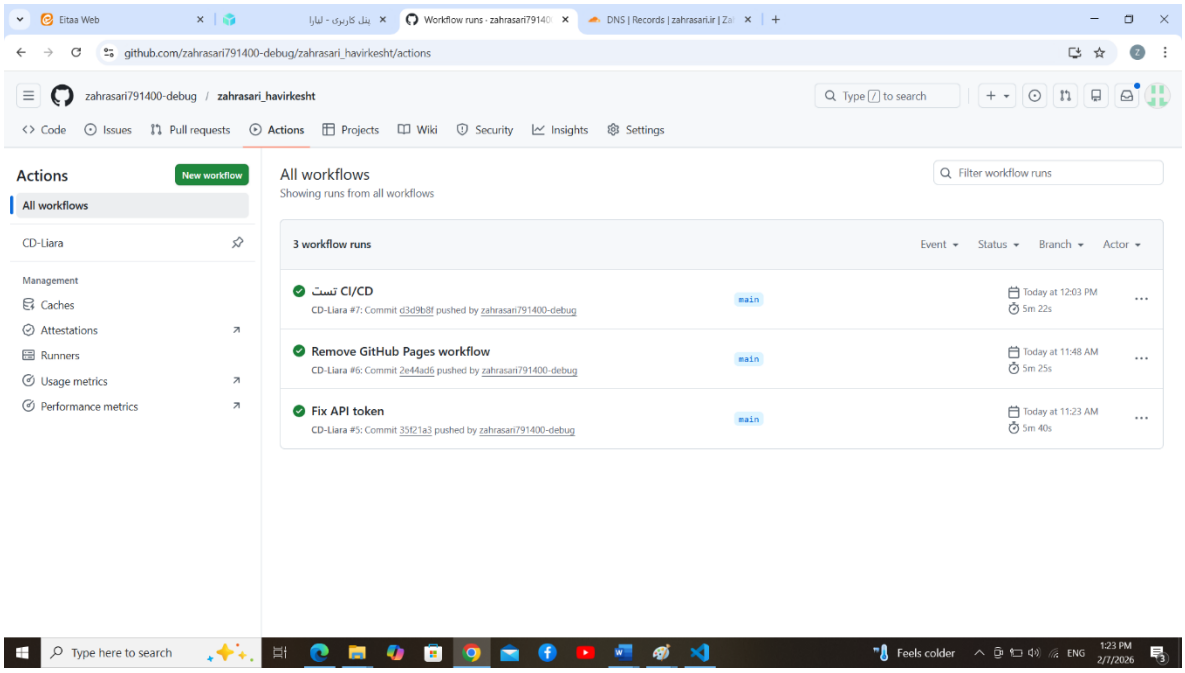
روی "ایجاد" کلیک کردم ۸ .توکن نمایش داده شد و آن را کپی کردم (توجه: این توکن فقط یک بار نمایش داده می‌شود)

قدم سوم: ساخت فایل Workflow حالا باید به گیت‌هاب می‌گفتم که چه کارهایی باید انجام دهد. برای این کار یک فایل ویژه ساختم :۱ .در پوشه اصلی پروژه، یک پوشه به نام github.ساختم ۲ .داخل آن، یک پوشه به نام workflows ساختم ۳ .داخلworkflows ، یک فایل به نام deploy.ymlساختم مسیر نهاییdeploy.yml/github/workflows. :

قدم چهارم: ارسال فایل workflow به گیت‌هاب " git commit -m افزودنgit add . CI/CDبا GitHub Actions" git push origin main تست :CI/CD برای اینکه مطمئن شوم همه چیز درست کار می‌کند، یک تست انجام دادم:

۱ .ایجاد تغییر: یک فایل جدید به نام test.txtساختم با محتوای "تست CI/CD" ۲ .ارسال تغییرات" git commit -m تست - CI/CD :git add test.txt اضافه کردن فایل تستی " git push origin main۳ .مشاهده در گیت‌هاب :وارد مخزن گیت‌هاب شدم روی تب "Actions"کلیک کردم دیدم که یک workflow جدید در حال اجرا است کنار نام commitیک دایره زرد بود که یعنی در حال اجراست

بررسی در لیارا :وارد پنل لیارا شدم دیدم که یک deployment جدید انجام شده تاریخ و ساعت با زمان push من مطابقت داشت


Eitaa Web
Workflow runs · zahrasari79140
DNS | Records | zahrasari.ir
github.com/zahrasari791400-debug/zahrasari_havirkesht/actions
zahrasari791400-debug / zahrasari_havirkesht
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
CD-Liara
Management
Caches
Attestations
Runners
Usage metrics
Performance metrics
All workflows
Showing runs from all workflows
Filter workflow runs
3 workflow runs
Event
Status
Branch
Actor
CI/CD تست
CD-Liara #7: Commit d3d9b8f pushed by zahrasari791400-debug
main
Today at 12:03 PM
5m 22s
Remove GitHub Pages workflow
CD-Liara #6: Commit 2e44ad6 pushed by zahrasari791400-debug
main
Today at 11:48 AM
5m 25s
Fix API token
CD-Liara #5: Commit 35f21a3 pushed by zahrasari791400-debug
main
Today at 11:23 AM
5m 40s
Type here to search
Feels colder
ENG
1:23 PM
2/7/2026

پایان