[redacted]

DAY 6 - DEPLOYMENT PREPARATION AND STAGING ENVIRONMENT SETUP

[redacted]

Date 21-01-25

DEPLOYMENT TESTING AND DOCUMENTATION

OBJECTIVE

The objective of Day 06 was to successfully deploy the "Bismillah Project" (Nike store project to a staging environment, ensure configuration of environment variables and document the testing and performance evaluation. This report encapsulates the achieved milestones, testing results and repository organization, highlighting a professional approach to the project's completion.

1 DEPLOYMENT DETAILS

01 Staging Environment

The Nike store is successfully deployed on Vercel, ensuring a fast, secure and scalable environment.

- URL: https://marketplace-nike-store.vercel.app/
- Hosting Provider: Vercel
- Staging Environment: Configured for testing before the production launch.

02 Environment Variables

Environment variables have been securely configured in Vercel to product sensitive

Dr. Saifuddin

Teacher's Signature

Date 21-01-25

03 **Staging Environment Setup**: Deployed the application to Vercel and verified successful deployment Checked content fetching from Sanity CMS.

04 **Staging Environment Testing**
Conducted Cypress functional tests, Postman API validation and Lighthouse performance tests. Ensure security with HTTPS, proper data handling

05 **Documentation Updates**: Created a READ.ME.md file all deployment instructions, configurations and test results. Included all reports in the GitHub repository.

STEPS FOR IMPLEMENTATION

Step 01: Hosting Platform Setup

Chosen Platform

* Vercel was selected for quick and easy deployment

Connect Respository

* Successfully connected the GitHub respository to Vercel for automatic deployments.
* Configured build settings and added the necessary scripts for deployment in the Vercel dashboard.

Step 02: Configure Environment Variables

Create .env.local File: Created the .env.local file to store sensitive data like API keys and token

information. Key variables include.

- SANITY-API-KEY: For secure communication with the Sanity backend.
- JWT-SECRET: For user authentication and secure sessions
- NEXT-PUBLIC-BASE-URL: Base URL for API Calls.

Upload Variables To Vercel:

* Uploaded the environment variables to Vercel using the platforms dashboard for secure handling.

Step 03: Deploy To Staging

Deploy Application

* Deployed the application to Vercel's staging environment for testing.

Validate Deployment

* Ensured the deployment build completed without errors.
* Verified that the application was loading correctly and all content was fetched properly from Sanity CMS.

Step 04: Staging Environment Testing

Testing Types

Functional Testing

* Product Listing: Ensured all products were listed correctly
* Product Details: Verified product details page

Dr. Saifuddin

Teacher's Signature

display correct information.

* User Profile: Checked User login, profile update & display.
* Cart Operations: Ensured products could be added, removed and quantities updated in the cart.
* Dynamic Routing: Verified that dynamic routing worked properly for product and category pages.

Performance Testing

* Used Lighthouse and GT metrix to analyze the performance Speed and responsiveness of the application.
* Ensure the application was optimized for various devices.

Security Testing

* Validated input fields to ensure they were protected from vulnerabilities such as SQL injection and other malicious attacks.
* Ensure HTTPS was enabled for secure communication between the client and server.

Test Case Reporting

Document all test cases in a CSV file with fields like Test Case ID, Description, Steps, Expected Results, Status and Remarks are attached in CSV file format.

Performance Testing

The performance test report generated by

Dr. Saifuddin

Teacher's Signature

Date 21-01-25

Lighthouse Tool: Lighthouse is an open-source, automated tool to help improve the quality of web pages. We can run it on any web page, public or requiring authentication. It has audits for performance, accessibility progressive web apps, SEO and more.

CONCLUSION

By the end of Day 06 we focused on setting up a staging environment for deployment, including configuring environment variables, testing functionality and updating documentation. This ensure a smooth and secure transition to the live platform, minimizing risks and enhancing readiness for production.

Nike Standard Issue Basketball Jersey

$2895

Nike Air Max 270

$13295

Nike Zoom Fly 5

$11295

Nike Air Force 1 PLT.AF.ORM

$8695

Nike Air Force 1 React

$13295

Air Jordan 1 Elevate Low

$11895

Nike Dri-FIT UV Hyverse

$2495

Nike Court Vision Low

$5695

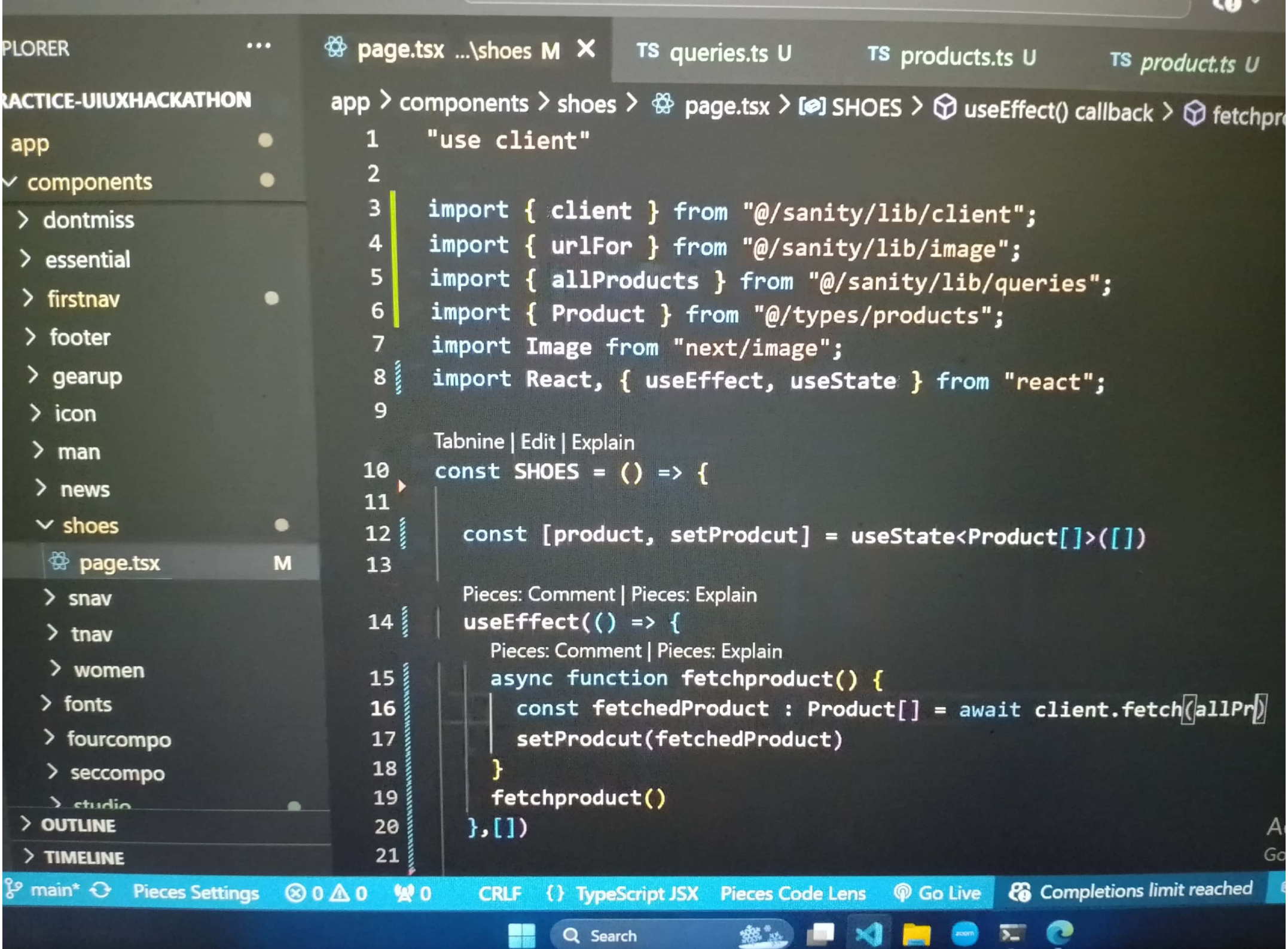
13295

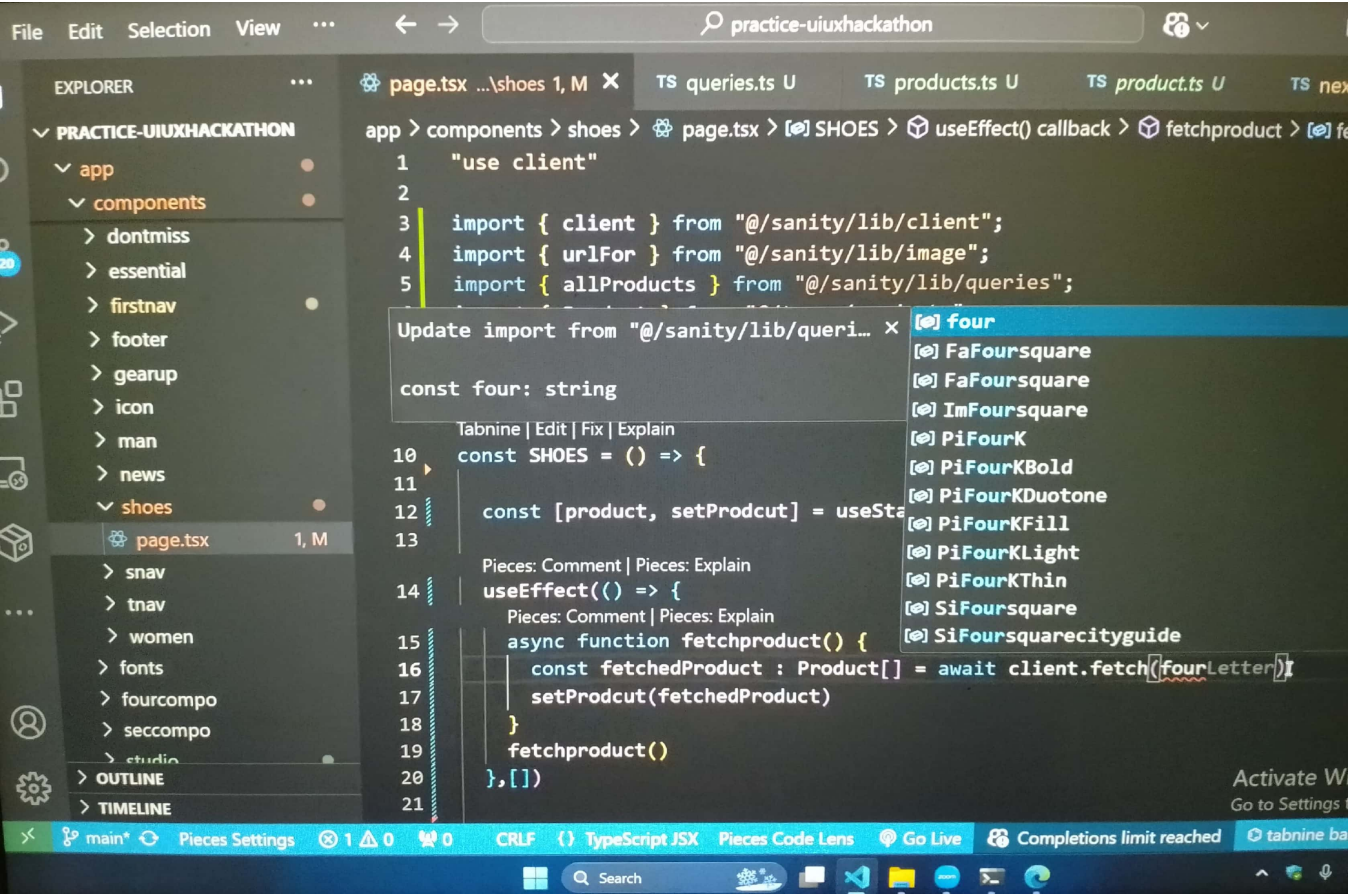
Nike Zoom Fly 5

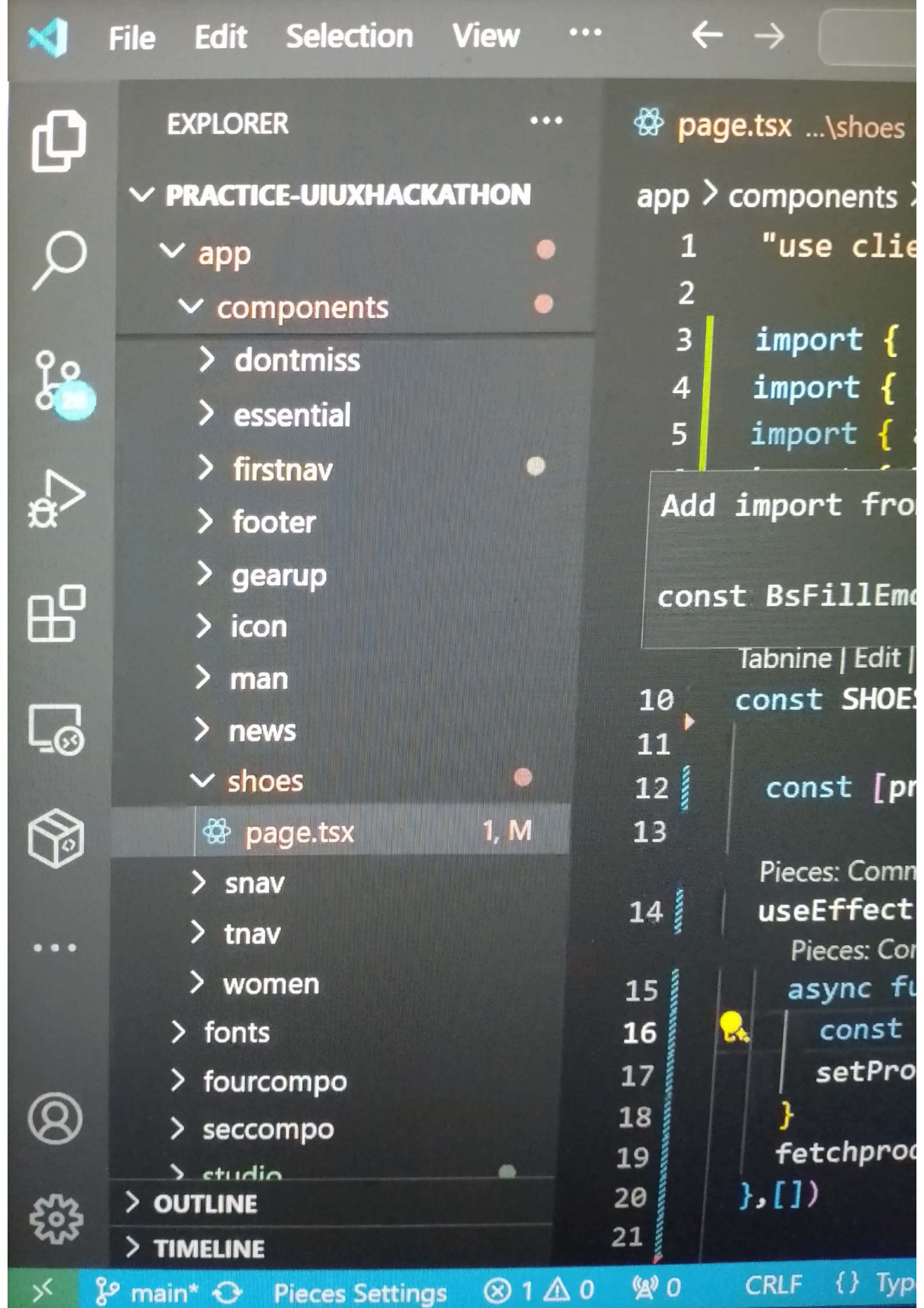
11295

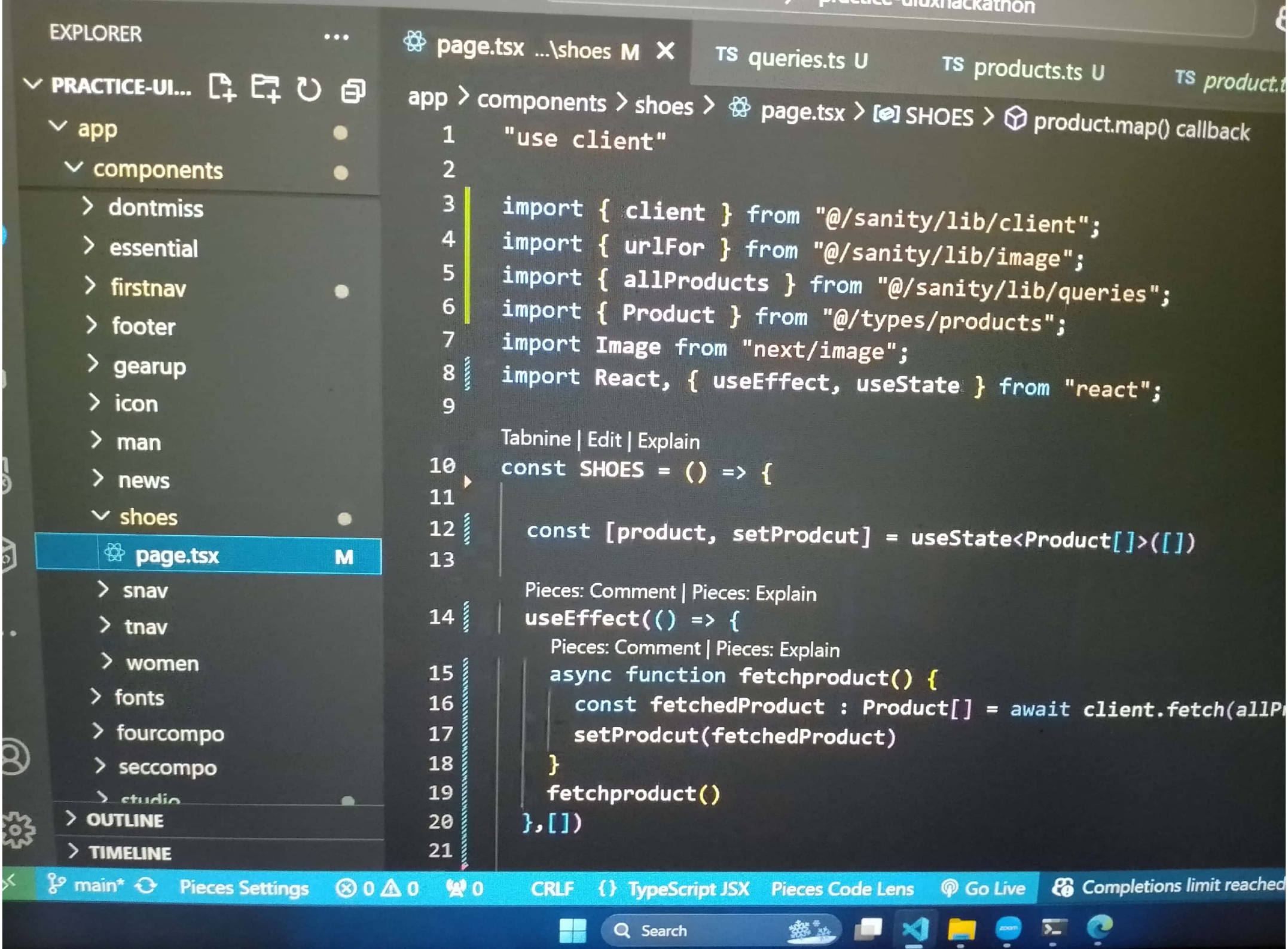
Nike Air Force 1 PLT.AF.ORM

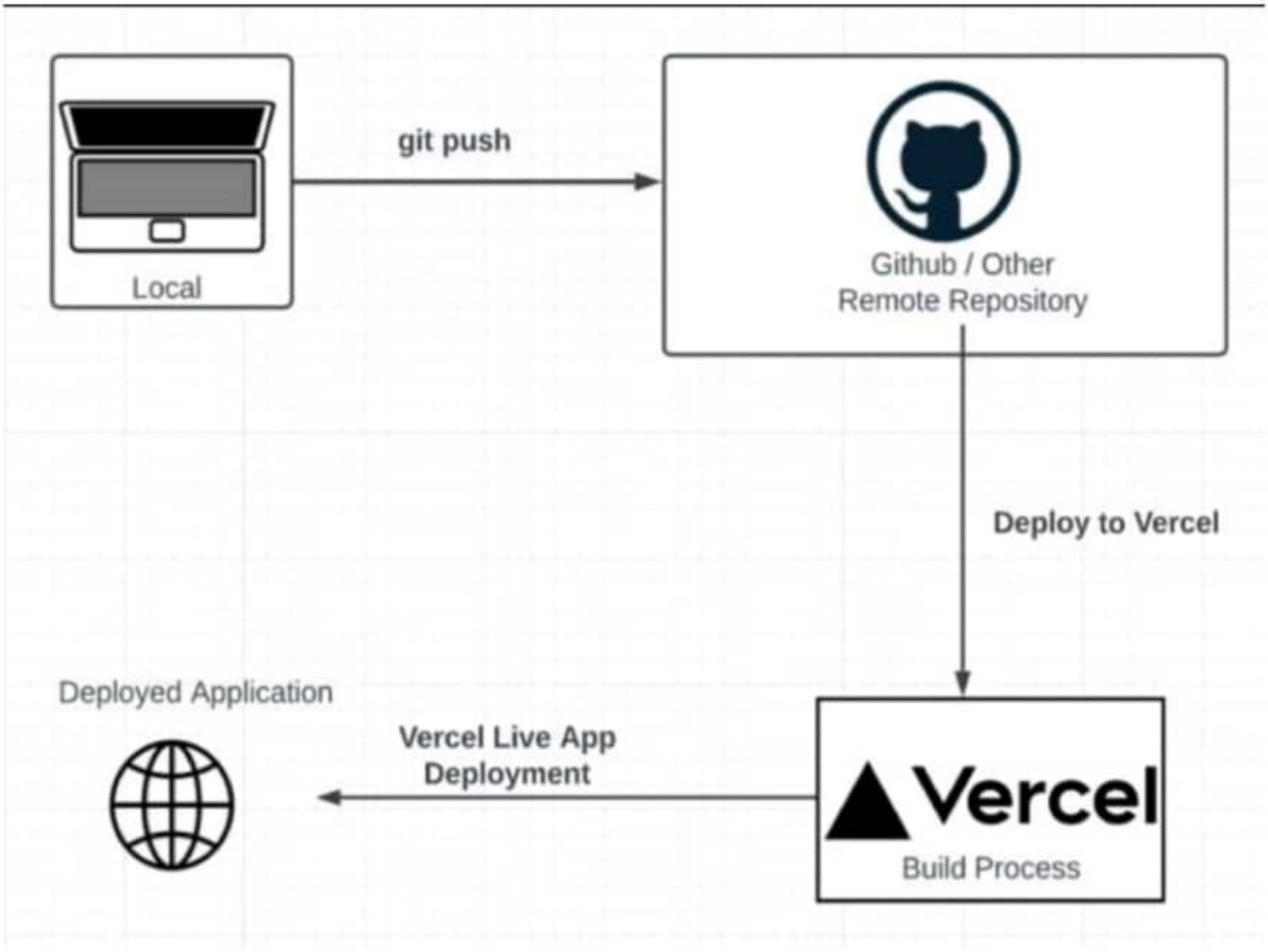
8695


page.tsx ...\shoes M
queries.ts U
products.ts U
product.ts U
app > components > shoes > page.tsx > SHOES > useEffect() callback > fetchpr
"use client"
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import { allProducts } from "@/sanity/lib/queries";
import { Product } from "@/types/products";
import Image from "next/image";
import React, { useEffect, useState } from "react";
Tabnine | Edit | Explain
const SHOES = () => {
const [product, setProdcut] = useState<Product[]>([])
Pieces: Comment | Pieces: Explain
useEffect(() => {
Pieces: Comment | Pieces: Explain
async function fetchproduct() {
const fetchedProduct : Product[] = await client.fetch(allPr)
setProdcut(fetchedProduct)
}
fetchproduct()
},[])
app
components
dontmiss
essential
firstnav
footer
gearup
icon
man
news
shoes
page.tsx M
snav
tnav
women
fonts
fourcompo
seccompo
OUTLINE
TIMELINE
main*
Pieces Settings
0 0
0
CRLF
TypeScript JSX
Pieces Code Lens
Go Live
Completions limit reached
Search


File Edit Selection View
practice-uiuxhackathon
EXPLORER
PRACTICE-UIUXHACKATHON
app
components
dontmiss
essential
firstnav
footer
gearup
icon
man
news
shoes
page.tsx 1, M
snav
tnav
women
fonts
fourcompo
seccompo
OUTLINE
TIMELINE
page.tsx ...\shoes 1, M
queries.ts U
products.ts U
product.ts U
app > components > shoes > page.tsx > SHOES > useEffect() callback > fetchproduct
"use client"
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import { allProducts } from "@/sanity/lib/queries";
Update import from "@/sanity/lib/queri…
const four: string
Tabnine | Edit | Fix | Explain
const SHOES = () => {
const [product, setProdcut] = useSta
Pieces: Comment | Pieces: Explain
useEffect(() => {
Pieces: Comment | Pieces: Explain
async function fetchproduct() {
const fetchedProduct : Product[] = await client.fetch(fourLetter)
setProdcut(fetchedProduct)
}
fetchproduct()
},[])
four
FaFoursquare
FaFoursquare
ImFoursquare
PiFourK
PiFourKBold
PiFourKDuotone
PiFourKFill
PiFourKLight
PiFourKThin
SiFoursquare
SiFoursquarecityguide
Activate W
Go to Settings
main*
Pieces Settings
1 0
0
CRLF
TypeScript JSX
Pieces Code Lens
Go Live
Completions limit reached
tabnine ba
Search


File Edit Selection View
EXPLORER
PRACTICE-UIUXHACKATHON
app
components
dontmiss
essential
firstnav
footer
gearup
icon
man
news
shoes
page.tsx 1, M
snav
tnav
women
fonts
fourcompo
seccompo
OUTLINE
TIMELINE
page.tsx ...\shoes
app > components
1 "use clie
2
3 import {
4 import {
5 import {
Add import fro
const BsFillEmo
Tabnine | Edit |
10 const SHOE
11
12 const [pr
13
Pieces: Comm
14 useEffect
Pieces: Co
15 async fu
16 const
17 setPro
18 }
19 fetchproc
20 },[])
21
main* Pieces Settings 1 0 0 CRLF {} Typ


EXPLORER
PRACTICE-UI...
app
components
dontmiss
essential
firstnav
footer
gearup
icon
man
news
shoes
page.tsx M
snav
tnav
women
fonts
fourcompo
seccompo
OUTLINE
TIMELINE
page.tsx ...\shoes M
queries.ts U
products.ts U
app > components > shoes > page.tsx > SHOES > product.map() callback
"use client"
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import { allProducts } from "@/sanity/lib/queries";
import { Product } from "@/types/products";
import Image from "next/image";
import React, { useEffect, useState } from "react";
Tabnine | Edit | Explain
const SHOES = () => {
const [product, setProdcut] = useState<Product[]>([])
Pieces: Comment | Pieces: Explain
useEffect(() => {
Pieces: Comment | Pieces: Explain
async function fetchproduct() {
const fetchedProduct : Product[] = await client.fetch(allP
setProdcut(fetchedProduct)
}
fetchproduct()
},[])
main*
Pieces Settings
0 0
0
CRLF
TypeScript JSX
Pieces Code Lens
Go Live
Completions limit reached
Search


Local
git push
Github / Other
Remote Repository
Deploy to Vercel
Deployed Application
Vercel Live App
Deployment
Vercel
Build Process

4. README.md Creation

Task:

Create a professional `README.md` file for the project.

What I Did:

Documented the entire process in a structured and well-organized README file.

How I Did It:

- Included **project overview, setup instructions, and deployment steps**.
- Added **guidelines for testing and environment configuration**.
- Documented **performance testing results** to highlight improvements.

Key Takeaways

✅ Successfully deployed the app to a staging environment on **Vercel**. ✅ Configured environment variables securely for safe deployment. ✅ Ran extensive **performance and security tests** to optimize the app. ✅ Created a **professional README.md** to document the entire development process.


اَلْحَمْدُ لِلّٰهِ رَبِّ الْعٰلَمِيْنَ