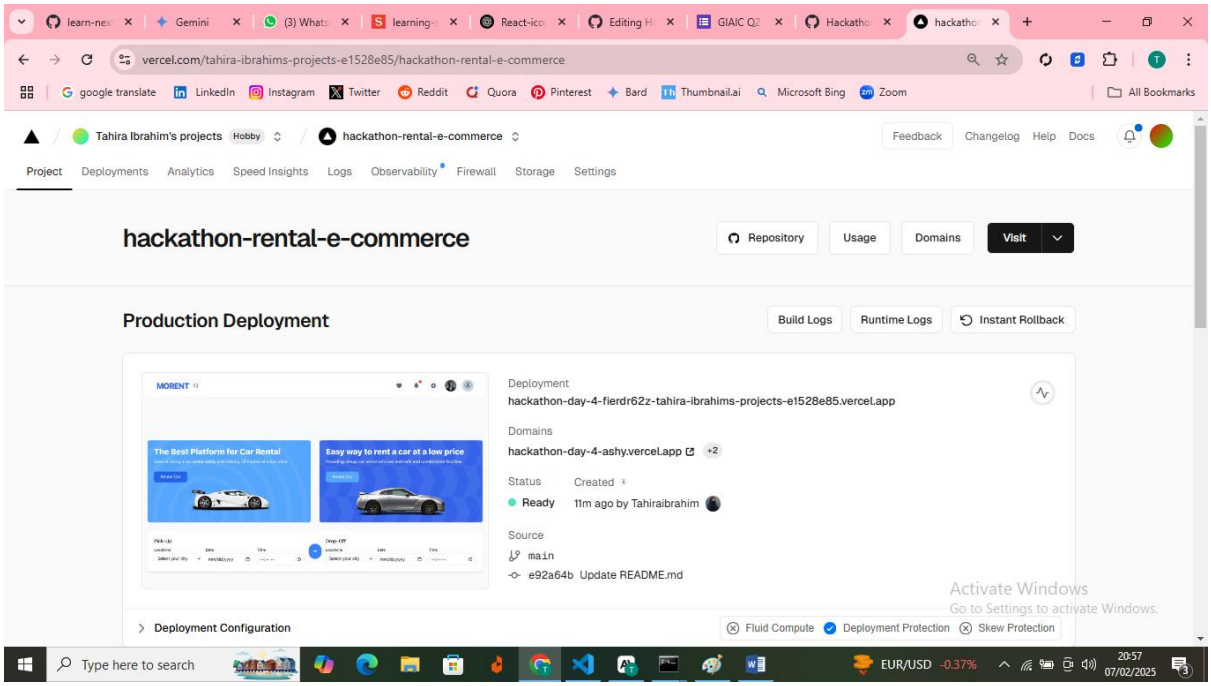
Hackathon Day 7

Live Deployment and Post-Launching

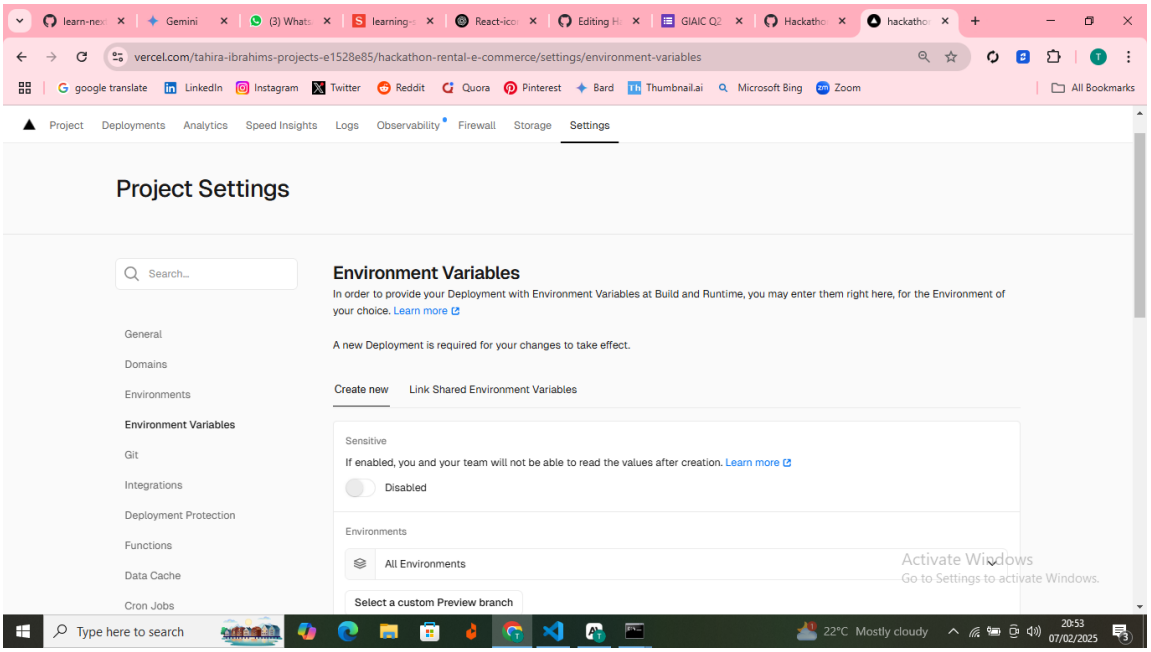
- On Day 7, we focus on the final steps of deployment and securing the application post-launch. This includes ensuring the app runs smoothly, is secure, and is optimized for performance.

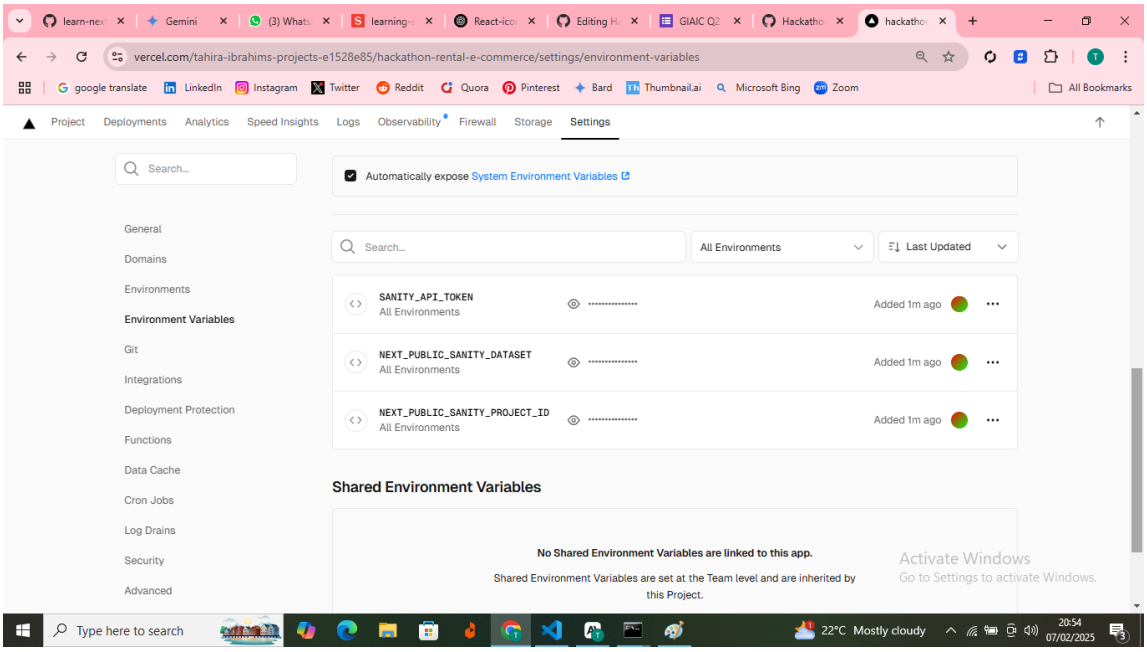


➢ Environment Variable Security and Configuration

When deploying an application to production, keeping sensitive information such as API keys, database credentials, and secrets safe is essential. To manage environment variables securely:

- Use. env files for local development and configure them for production.
- In production environments (like Vercel), store sensitive values using their respective secret management tools.
- Ensure the environment variables are not exposed in the codebase or version control systems (e.g., GitHub).





➢ . SSL and HTTPS Enable

Securing your app with SSL (Secure Socket Layer) ensures encrypted data transfer between the client and server. Enabling HTTPS on your website will protect it from man-in-the-middle attacks and increase trustworthiness. Here's how to enable it:

- Use free SSL certificates from services like Let’s Encrypt or paid ones for greater support.
- Configure your server (e.g., Nginx or Apache) or hosting service (e.g., Vercel or Netlify) to enforce HTTPS.

- Redirect all HTTP traffic to HTTPS to ensure secure communication.

https://hackathon-day-4-ashy.vercel.app/

➢ Codebase Security and Repository Management

After deploying, it is important to continue managing the codebase securely:

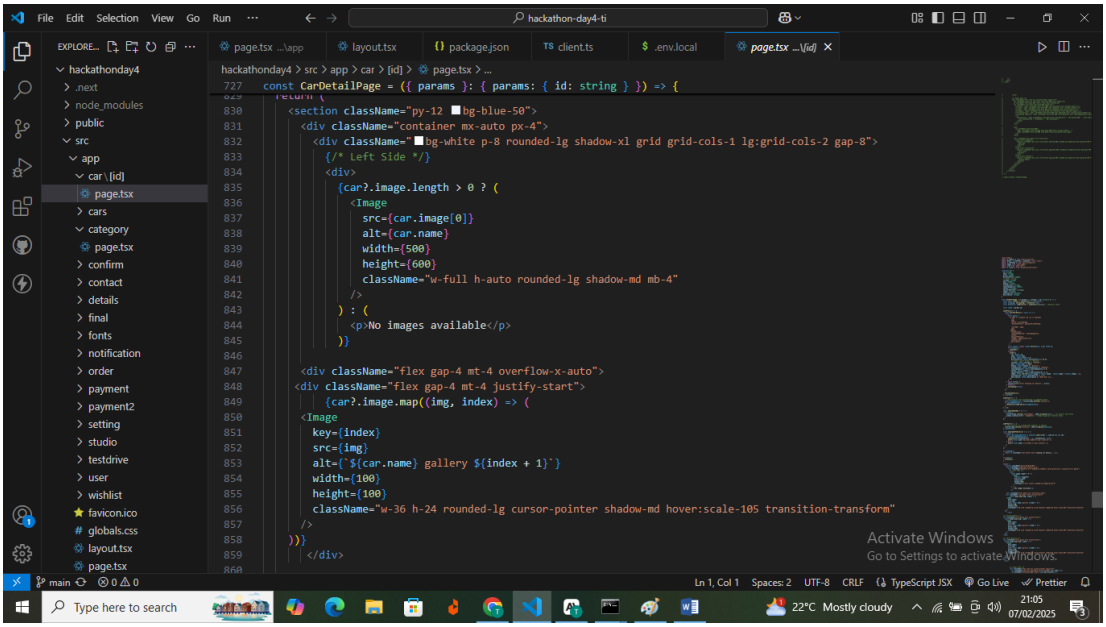
- Use Git for version control and maintain clear commit messages.
- Implement code review practices to detect vulnerabilities early.
- Avoid storing sensitive data (e.g., passwords) directly in the repository; use environment variables instead.
- Set up branch protection rules for collaborative development.

➢ Performance Optimization

Improving performance is critical for user satisfaction. Key techniques include:

- **Code Splitting**: Break down large JavaScript files into smaller chunks to load only what is needed.
- **Lazy Loading**: Load images, videos, and content only when they are in the viewport.
- **Image Optimization**: Use modern formats (e.g., Web) and compress images to reduce load times.
- **Magnification and Compression**: Minify CSS, JS, and HTML files, and use gzip or Brotli for compression.

  <image src" /path/to/image.jpg" width = {500} height = {500} alt = "Alt text"/>



➢ Monitoring and Analytics Tools

Post-launch, it's important to track your app's performance and user behaviour. Use these tools:

- **Versal Analytics**: To track user interactions, session durations, and conversions.
- **Sentry**: For monitoring application errors in real-time.
- **New Relic or Datadog**: To monitor server-side performance and detect issues early.

Use tools like Sentry

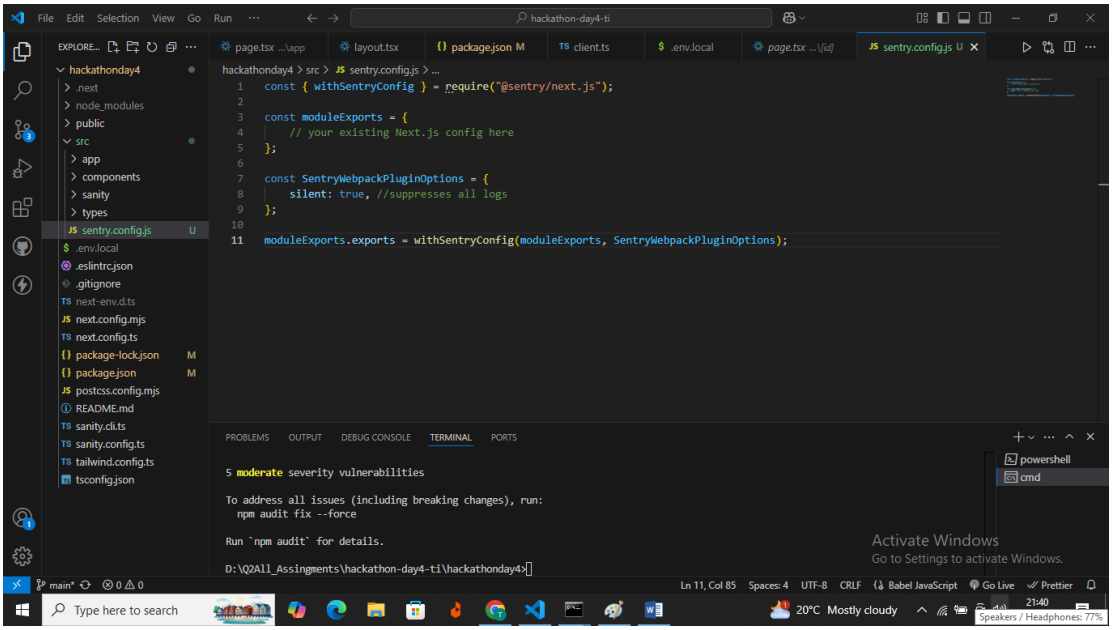
Npm install @sentry/nextjs

```
const {withSentryConfig } = require("@sentry/next.js");

const moduleExports = {
    // your existing Next.js config here
};

const SentryWebpackPluginOptions = {
    silent: true, //suppresses all logs
};

moduleExports.exports = withSentryConfig (moduleExports,
SentryWebpackPluginOptions);
```



➢ Backup and Disaster Recovery

To ensure that the application remains functional in case of issues, set up:

- **Regular Backups**: Automatically back up the database and essential files (e.g., nightly backups).
- **Disaster Recovery Plan**: Have a strategy in place to quickly restore services in case of failures. This includes having a replica of the production environment and data backups stored securely in multiple locations.

Conclusion

As we wrap up **Hackathon Day 7**, the focus has been on ensuring a smooth and secure deployment, optimizing performance, and setting up monitoring for long-term success. Deploying an application is not just about making it live; it requires security configurations, HTTPS enforcement, repository management, performance tuning, and real-time monitoring to maintain a seamless user experience.

Additionally, having backup and disaster recovery strategies ensures that unforeseen issues do not disrupt the service. By implementing these best practices, we ensure that our application is secure, efficient, and scalable, setting a strong foundation for future improvements and growth.

With this, my hackathon journey moves towards its final stages, and our project is now live and ready for real-world users!