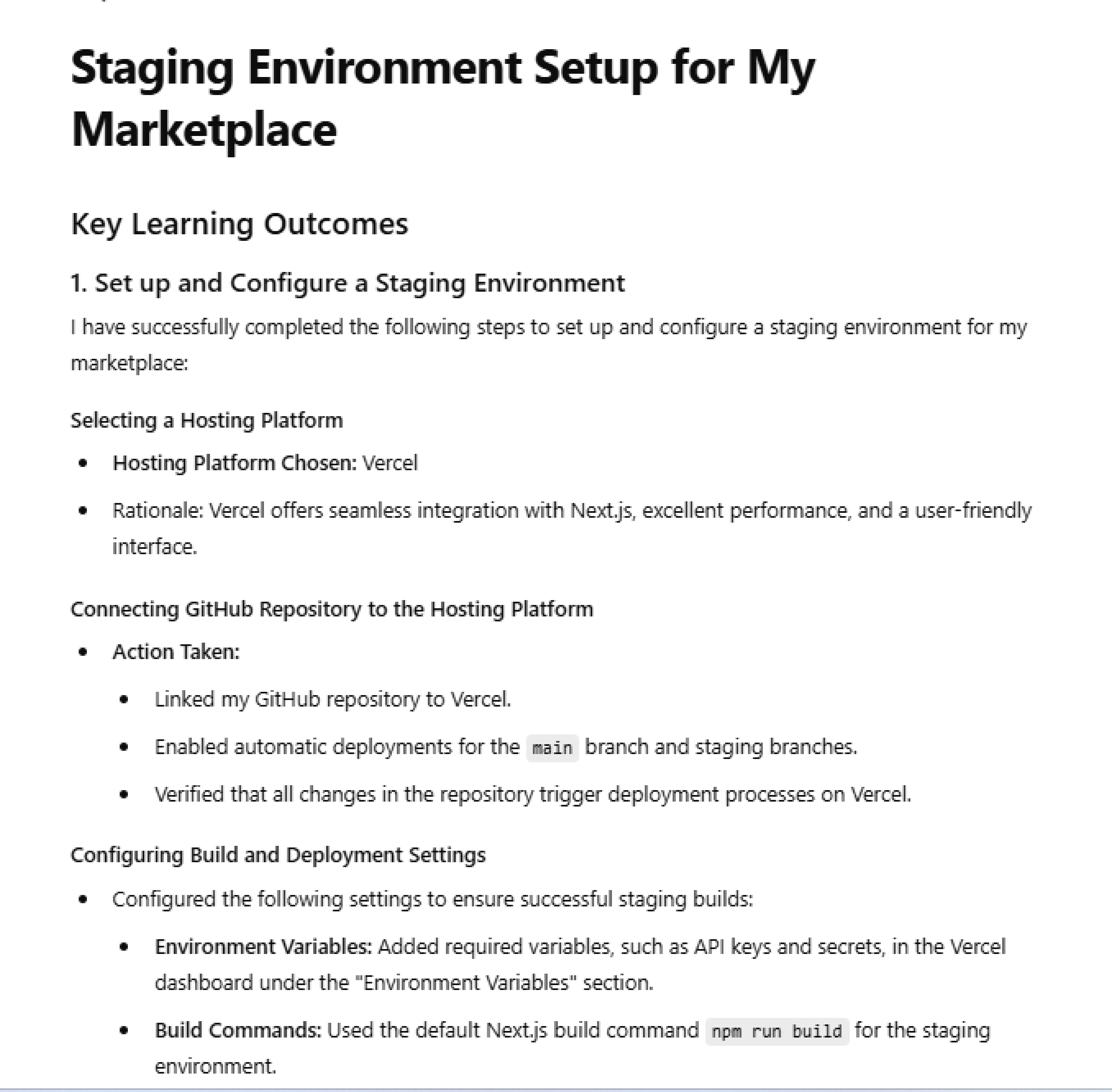


# Staging Environment Setup for My Marketplace

## Key Learning Outcomes

### 1. Set up and Configure a Staging Environment

I have successfully completed the following steps to set up and configure a staging environment for my marketplace:

#### Selecting a Hosting Platform

- **Hosting Platform Chosen:** Vercel
- Rationale: Vercel offers seamless integration with Next.js, excellent performance, and a user-friendly interface.

#### Connecting GitHub Repository to the Hosting Platform

- **Action Taken:**
  - Linked my GitHub repository to Vercel.
  - Enabled automatic deployments for the `main` branch and staging branches.
  - Verified that all changes in the repository trigger deployment processes on Vercel.

#### Configuring Build and Deployment Settings

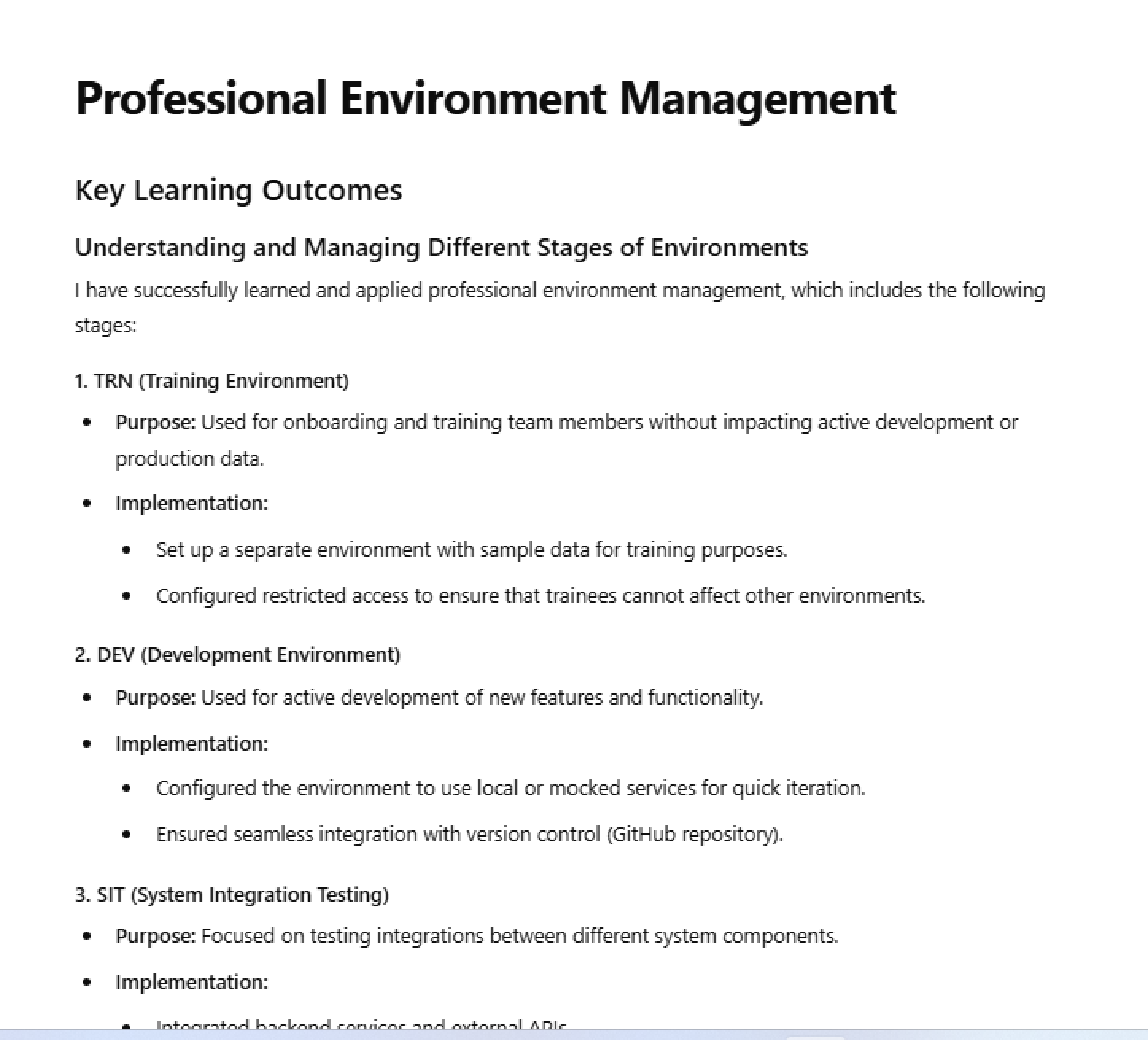
- Configured the following settings to ensure successful staging builds:
  - **Environment Variables:** Added required variables, such as API keys and secrets, in the Vercel dashboard under the "Environment Variables" section.
  - **Build Commands:** Used the default Next.js build command `npm run build` for the staging environment.

environment.

  - **Output Directory:** Ensured that the `.next` directory is used for deploying the build.
  - **Branch Previews:** Enabled branch previews for staging to test features before merging them into the production branch.

### Testing and Validation

- Verified that the staging environment:
  - Reflects the latest changes pushed to the `staging` branch.
  - Is accessible and performs as expected on various devices and browsers.



# Professional Environment Management

## Key Learning Outcomes

### Understanding and Managing Different Stages of Environments

I have successfully learned and applied professional environment management, which includes the following stages:

1. TRN (Training Environment)

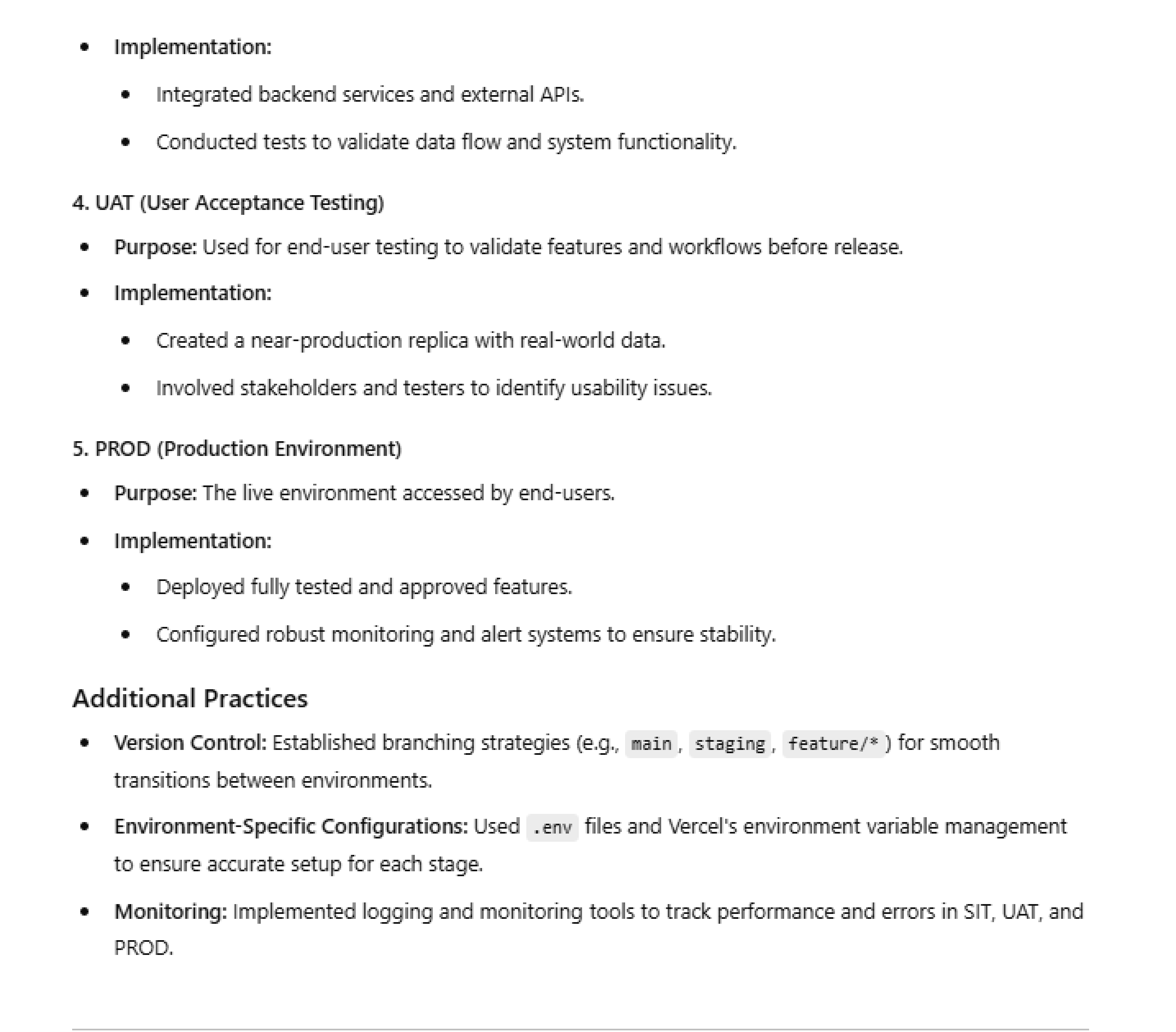
- **Purpose:** Used for onboarding and training team members without impacting active development or production data.
- **Implementation:**
  - Set up a separate environment with sample data for training purposes.
  - Configured restricted access to ensure that trainees cannot affect other environments.

2. DEV (Development Environment)

- **Purpose:** Used for active development of new features and functionality.
- **Implementation:**
  - Configured the environment to use local or mocked services for quick iteration.
  - Ensured seamless integration with version control (GitHub repository).

3. SIT (System Integration Testing)

- **Purpose:** Focused on testing integrations between different system components.
- **Implementation:**
  - Integrated backend services and external APIs.



- **Implementation:**
  - Integrated backend services and external APIs.
  - Conducted tests to validate data flow and system functionality.

### 4. UAT (User Acceptance Testing)

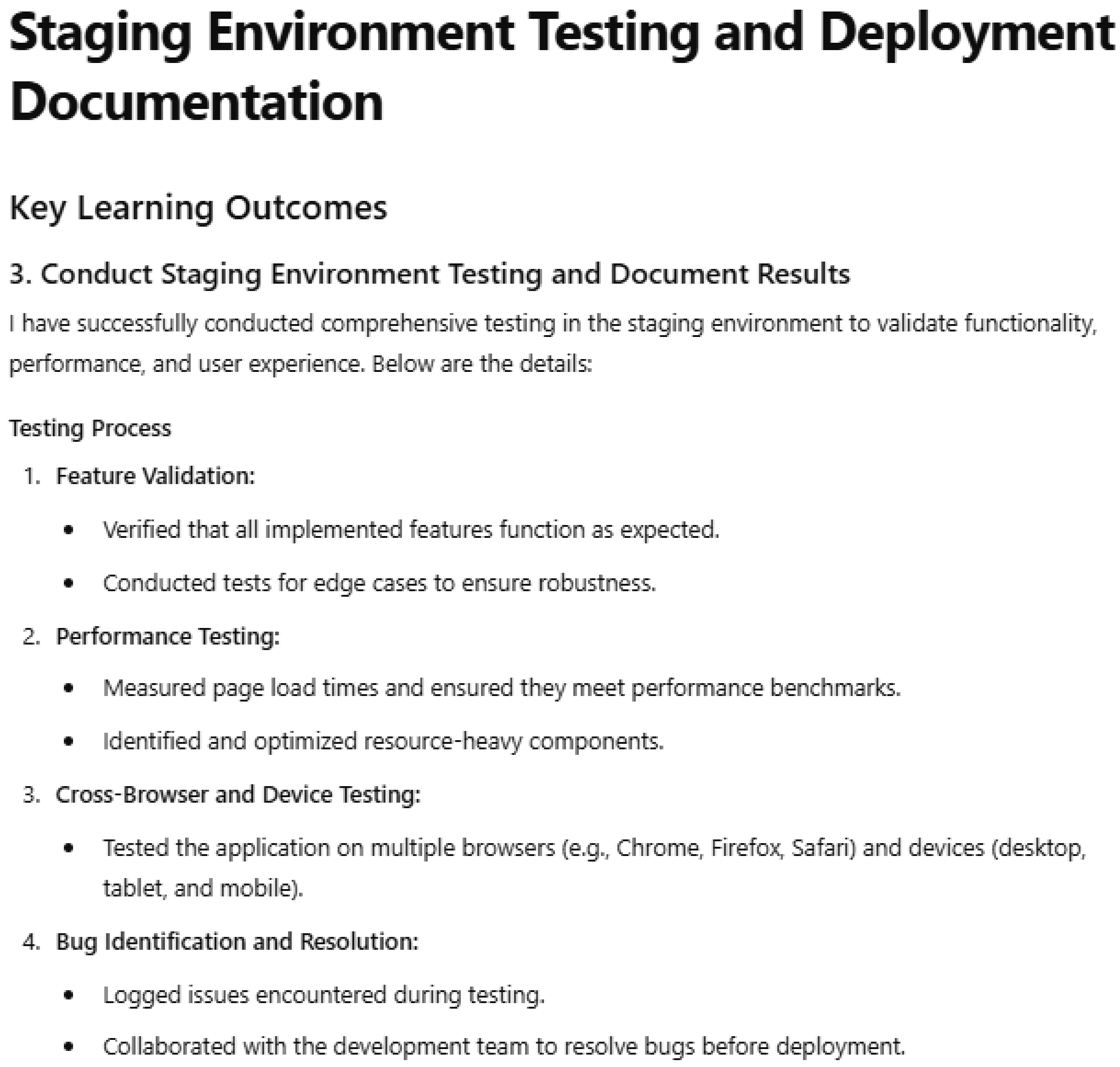
- **Purpose:** Used for end-user testing to validate features and workflows before release.
- **Implementation:**
  - Created a near-production replica with real-world data.
  - Involved stakeholders and testers to identify usability issues.

### 5. PROD (Production Environment)

- **Purpose:** The live environment accessed by end-users.
- **Implementation:**
  - Deployed fully tested and approved features.
  - Configured robust monitoring and alert systems to ensure stability.

## Additional Practices

- **Version Control:** Established branching strategies (e.g., `main`, `staging`, `feature/*`) for smooth transitions between environments.
- **Environment-Specific Configurations:** Used `.env` files and Vercel's environment variable management to ensure accurate setup for each stage.
- **Monitoring:** Implemented logging and monitoring tools to track performance and errors in SIT, UAT, and PROD.



# Staging Environment Testing and Deployment Documentation

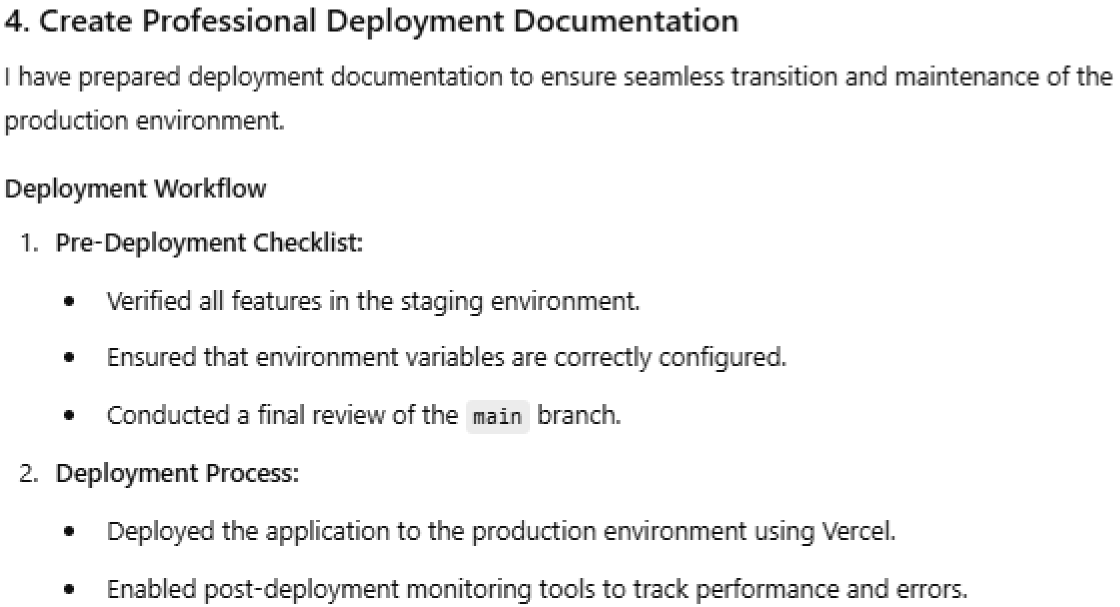
## Key Learning Outcomes

### 3. Conduct Staging Environment Testing and Document Results

I have successfully conducted comprehensive testing in the staging environment to validate functionality, performance, and user experience. Below are the details:

Testing Process

1. **Feature Validation:**
   - Verified that all implemented features function as expected.
   - Conducted tests for edge cases to ensure robustness.
2. **Performance Testing:**
   - Measured page load times and ensured they meet performance benchmarks.
   - Identified and optimized resource-heavy components.
3. **Cross-Browser and Device Testing:**
   - Tested the application on multiple browsers (e.g., Chrome, Firefox, Safari) and devices (desktop, tablet, and mobile).
4. **Bug Identification and Resolution:**
   - Logged issues encountered during testing.
   - Collaborated with the development team to resolve bugs before deployment.

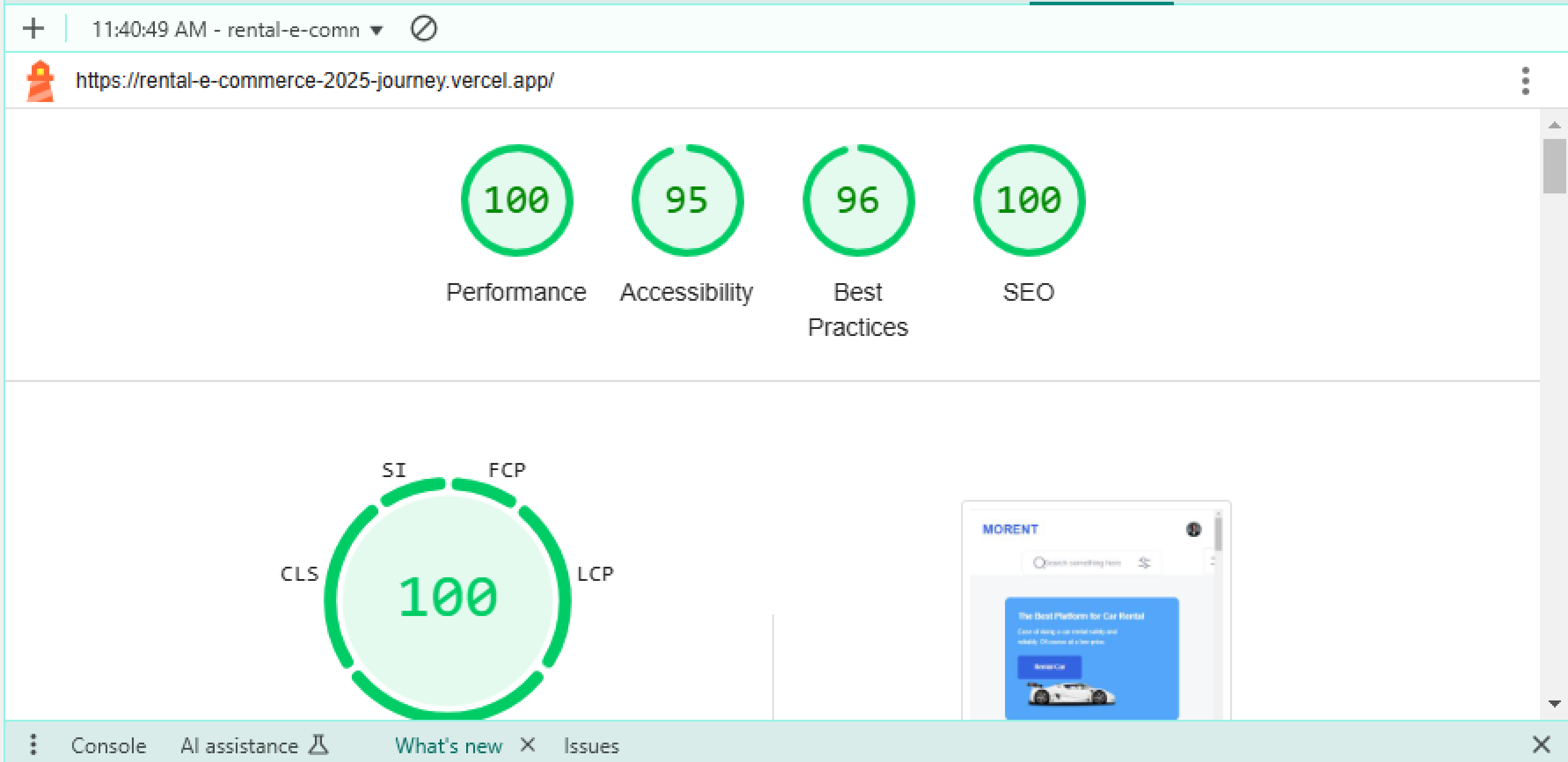


## 4. Create Professional Deployment Documentation

I have prepared deployment documentation to ensure seamless transition and maintenance of the production environment.

### Deployment Workflow

1. **Pre-Deployment Checklist:**
   - Verified all features in the staging environment.
   - Ensured that environment variables are correctly configured.
   - Conducted a final review of the `main` branch.
2. **Deployment Process:**
   - Deployed the application to the production environment using Vercel.
   - Enabled post-deployment monitoring tools to track performance and errors.



11:40:49 AM - rental-e-comn

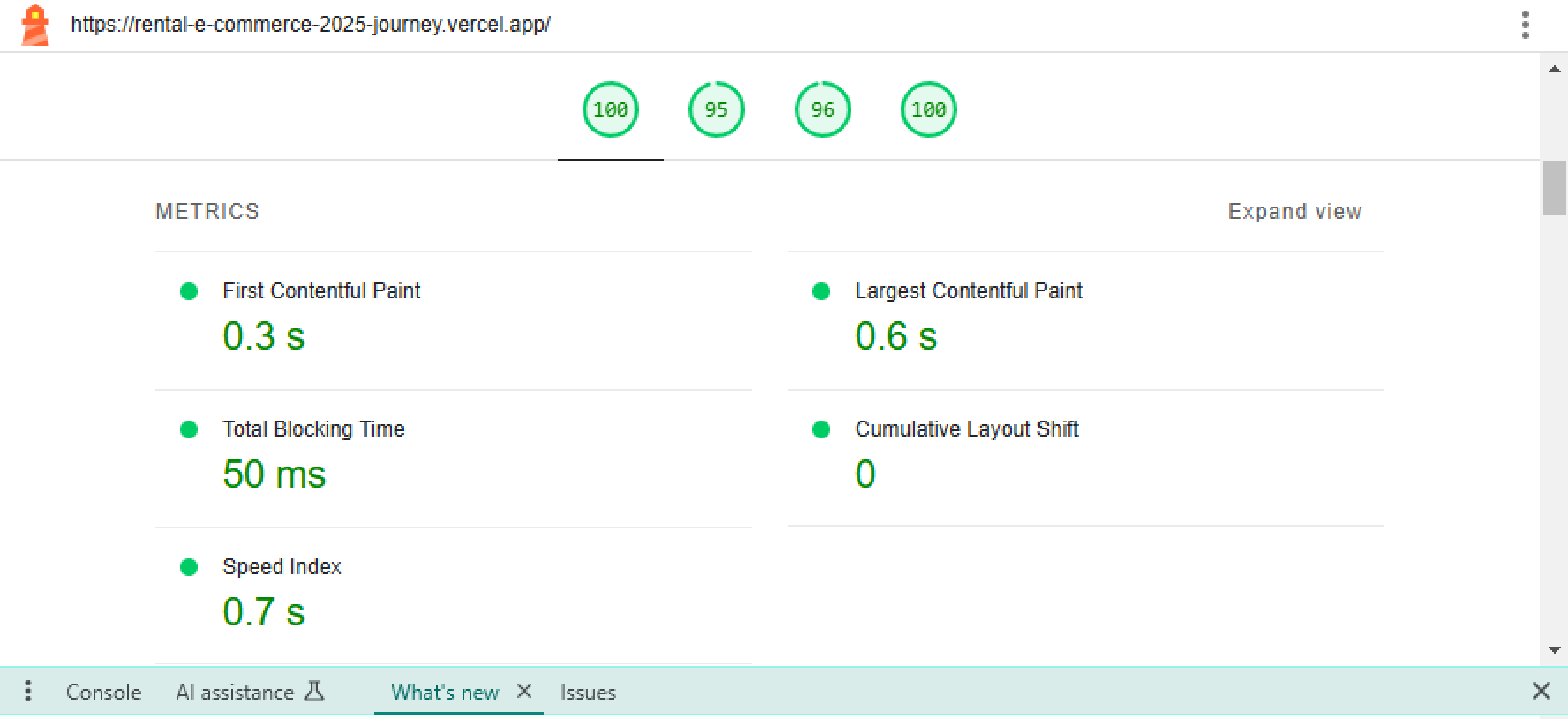
https://rental-e-commerce-2025-journey.vercel.app/

100 Performance

95 Accessibility

96 Best Practices

100 SEO



https://rental-e-commerce-2025-journey.vercel.app/

METRICS

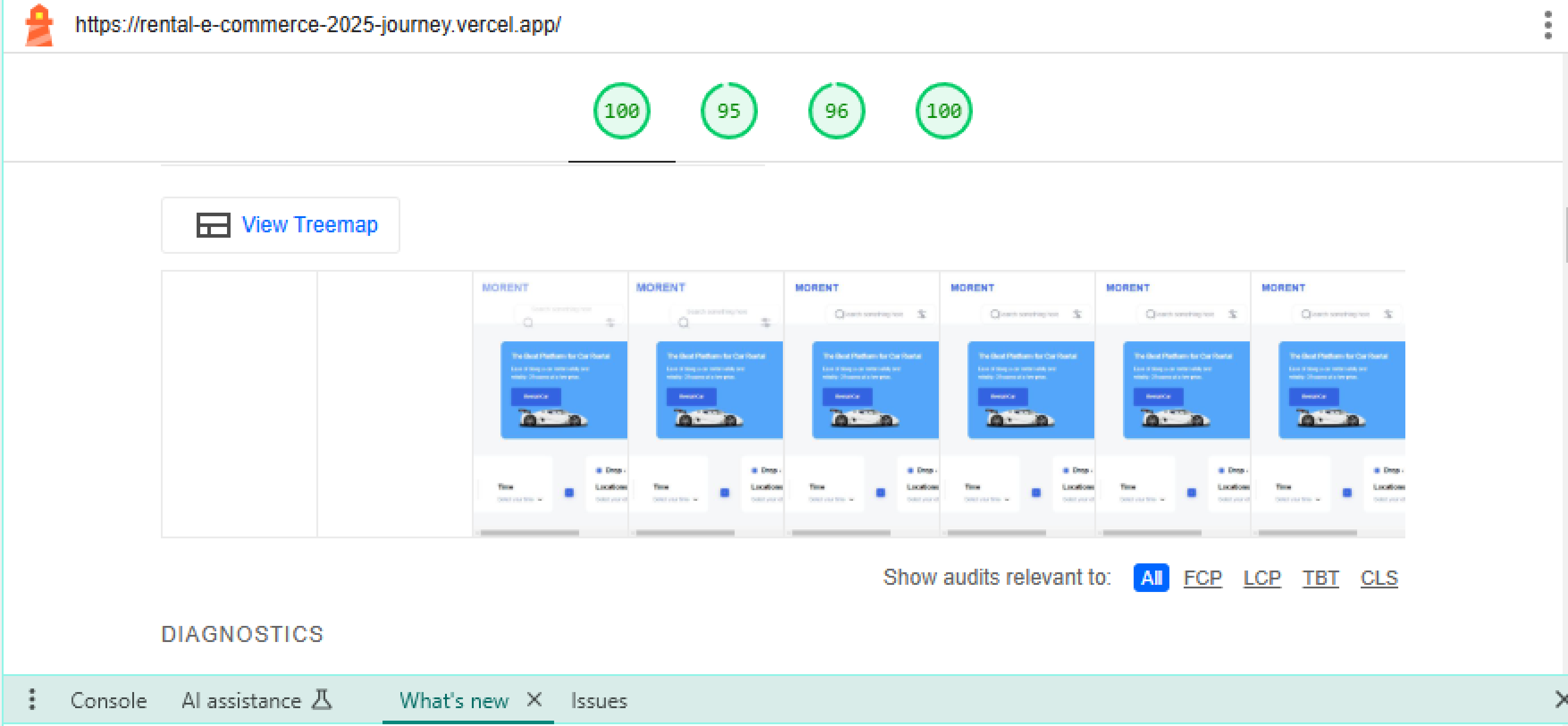
● First Contentful Paint
0.3 s

● Largest Contentful Paint
0.6 s

● Total Blocking Time
50 ms

● Cumulative Layout Shift
0

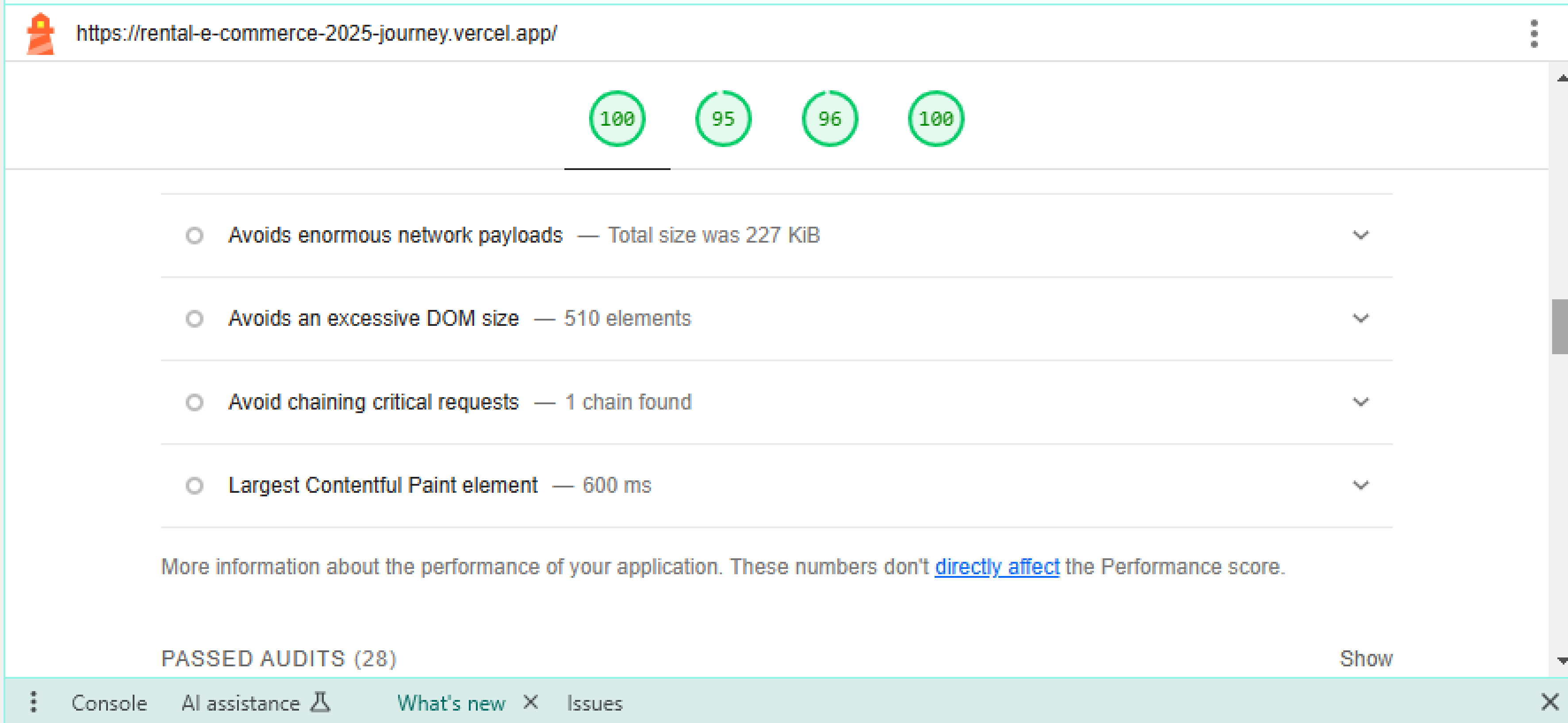
● Speed Index
0.7 s



https://rental-e-commerce-2025-journey.vercel.app/

100 95 96 100

DIAGNOSTICS



https://rental-e-commerce-2025-journey.vercel.app/

○ Avoids enormous network payloads — Total size was 227 KiB

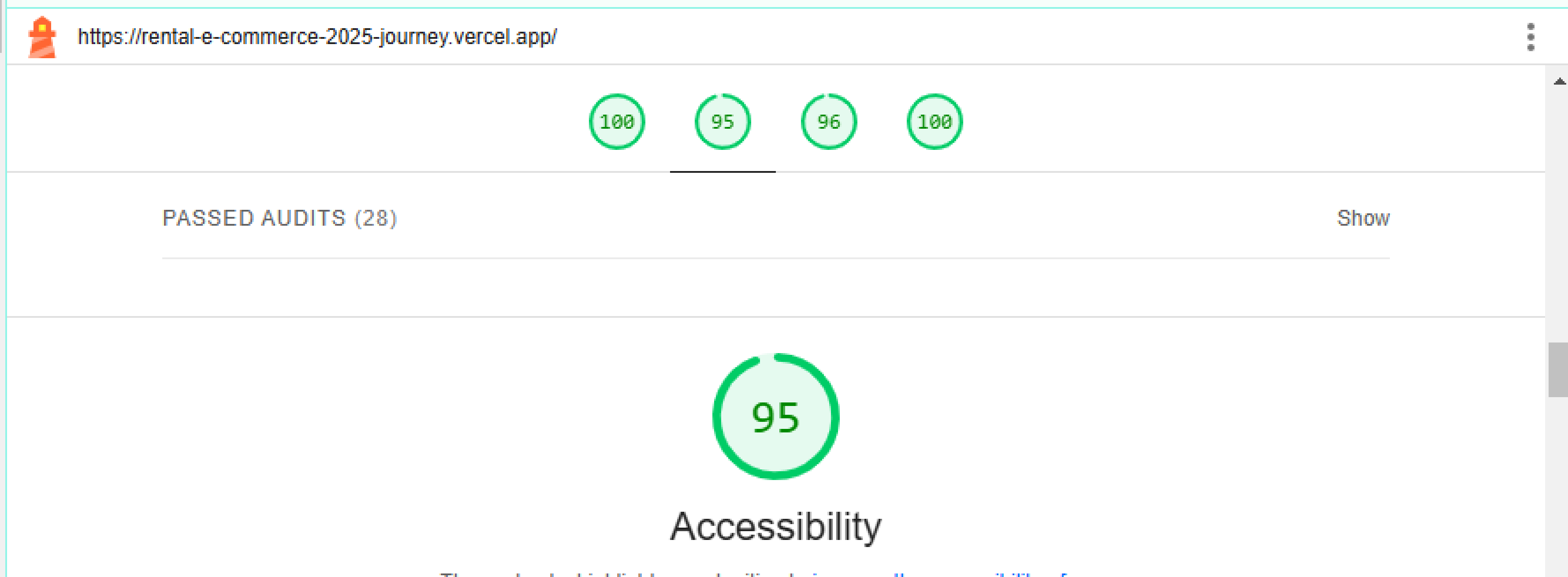
○ Avoids an excessive DOM size — 510 elements

○ Avoid chaining critical requests — 1 chain found

○ Largest Contentful Paint element — 600 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

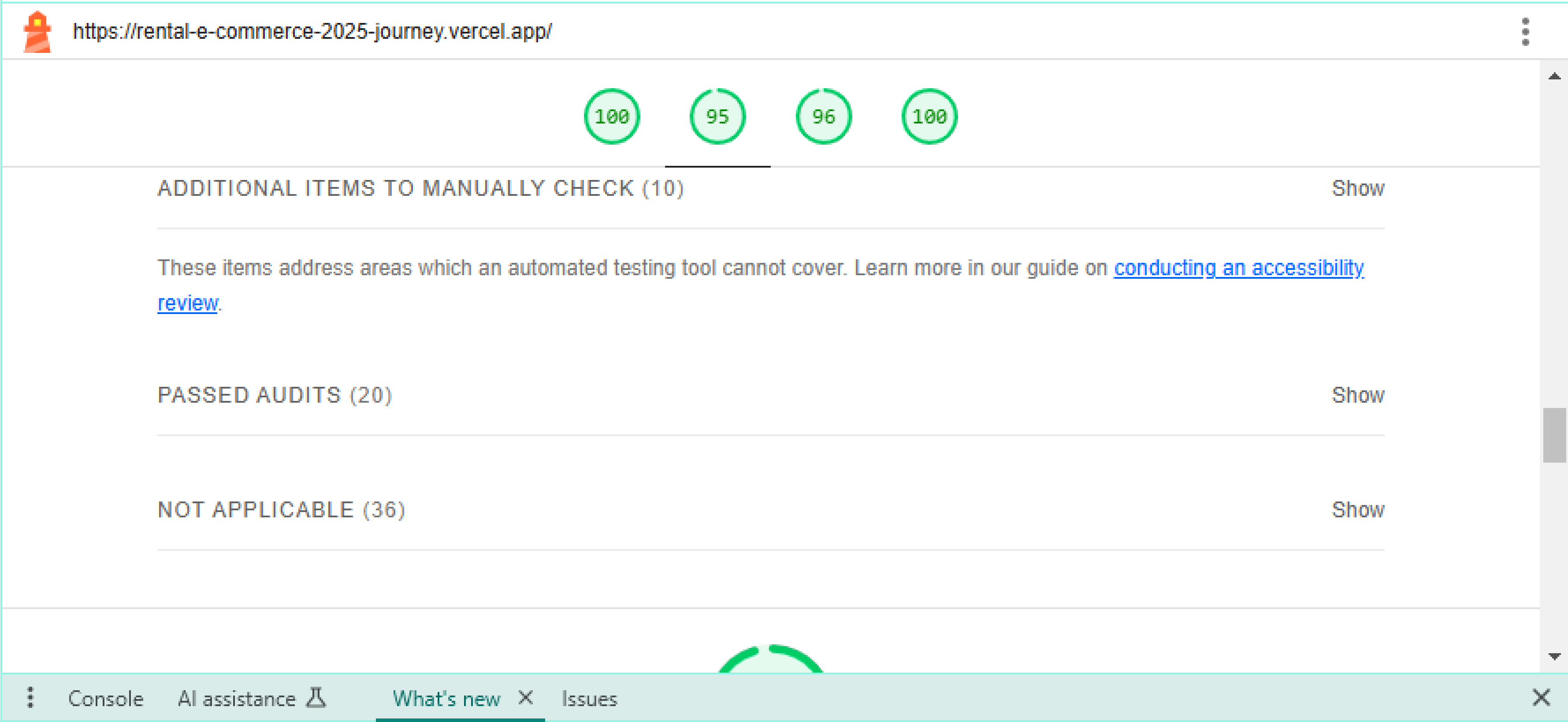
PASSED AUDITS (28) Show



https://rental-e-commerce-2025-journey.vercel.app/

PASSED AUDITS (28)

Accessibility



https://rental-e-commerce-2025-journey.vercel.app/

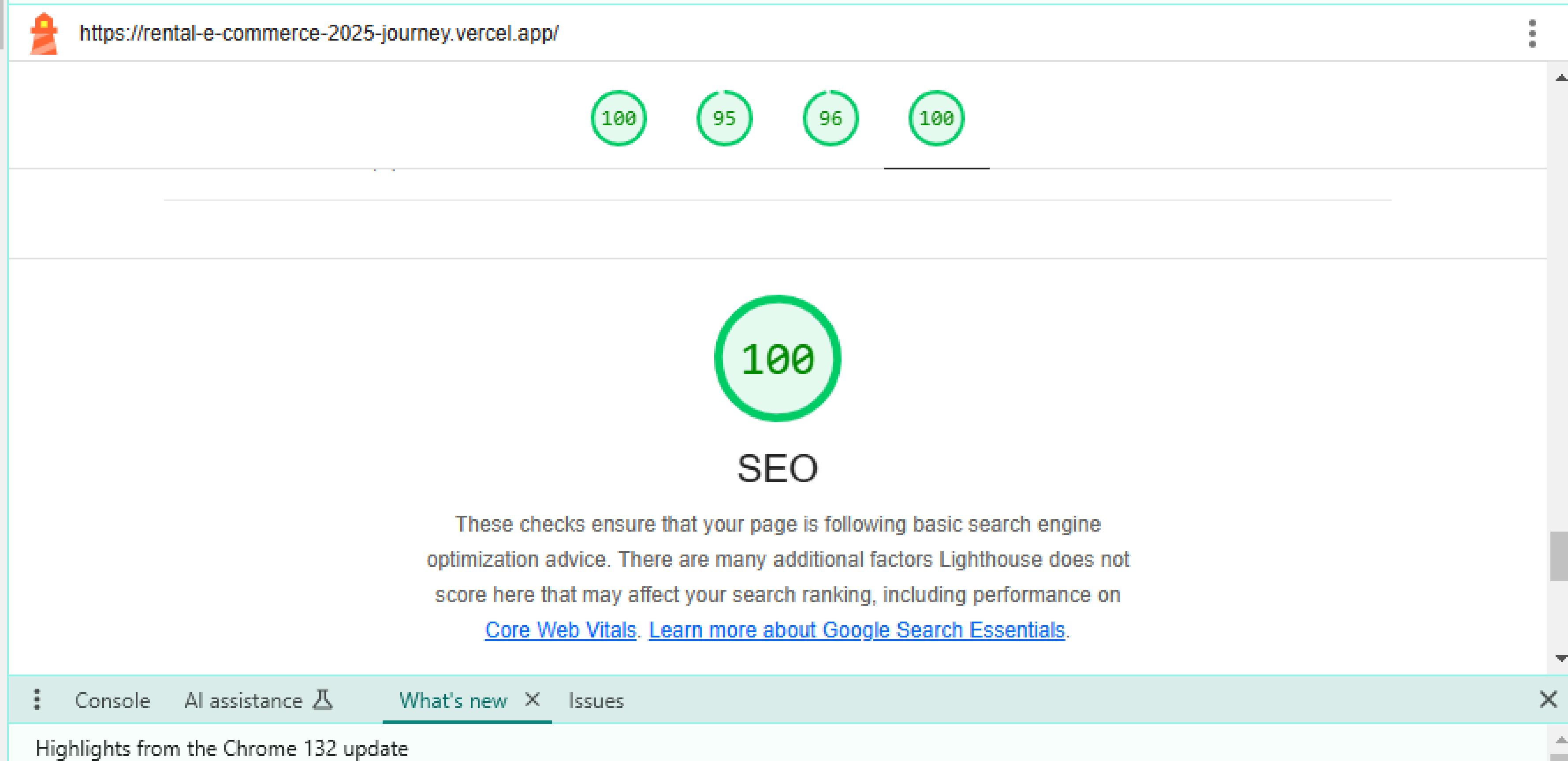
100 95 96 100

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (20) Show

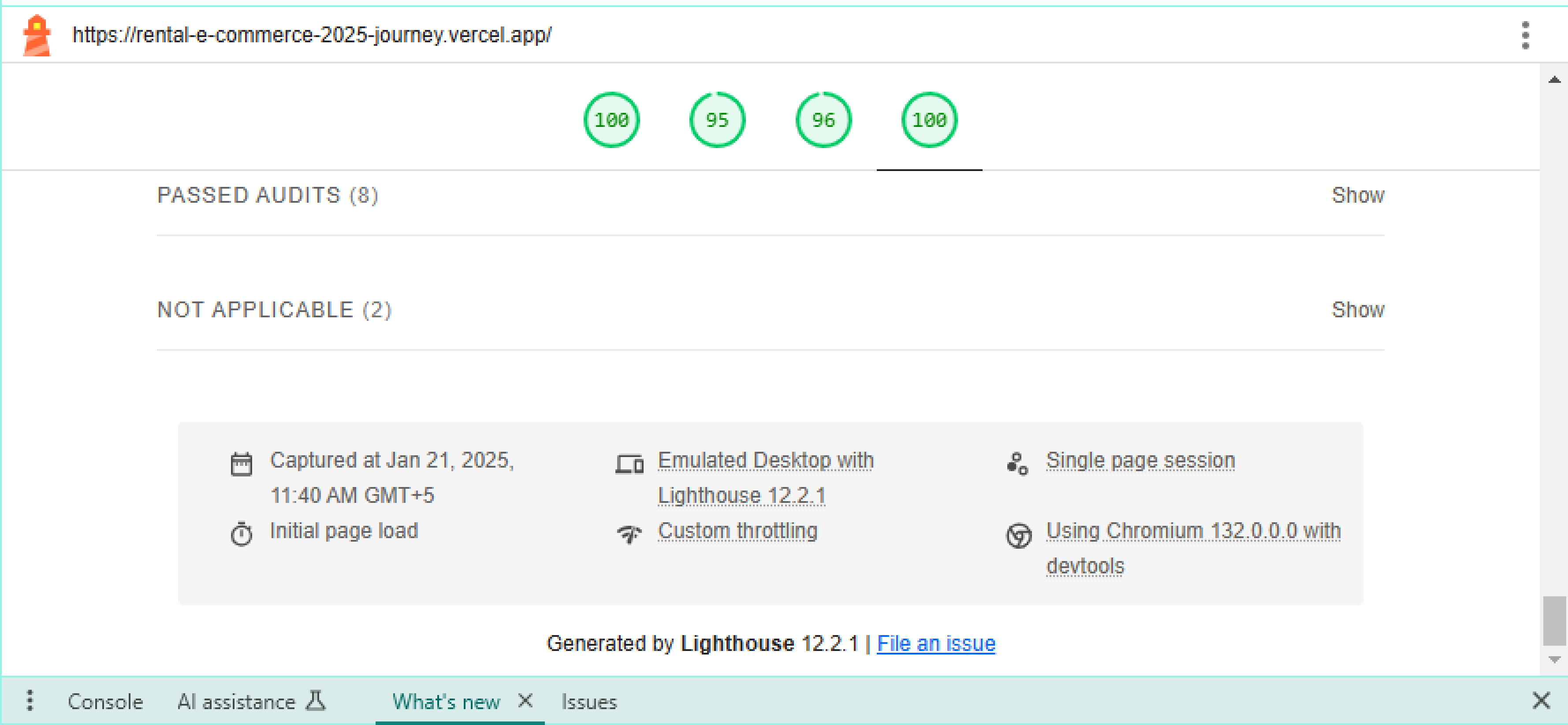
NOT APPLICABLE (36) Show



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.

Highlights from the Chrome 132 update



PASSED AUDITS (8)

NOT APPLICABLE (2)

Captured at Jan 21, 2025, 11:40 AM GMT+5

Emulated Desktop with Lighthouse 12.2.1

Single page session

Initial page load

Custom throttling

Using Chromium 132.0.0.0 with devtools

Generated by **Lighthouse** 12.2.1 | File an issue

| Test ID | Test Case Description | Expected Result | Actual Result | Status | Remarks |
|---|---|---|---|---|---|
| TST-001 | Homepage Load Test | Homepage should load within 2 seconds | Homepage loaded within 2 seconds | Passed | Displayed Properly |
| TST-002 | Car Display Test | All cars should be displayed correctly on the homepage | All cars displayed correctly | Passed | No Issues found |
| TST-003 | Car Details Test | Car details should display upon selection | Car details displayed correctly | Passed | Showing Correctly |
| TST-004 | Checkout Page Test | Checkout process should complete without errors | Checkout completed successfully | Passed | Successfully |
| TST-005 | API Integration Test | API should fetch and display data correctly | API fetched and displayed data as expected | Passed | Working Properly |
| TST-006 | Navigation Test | All navigation links should be functional | All navigation links work correctly | Passed | Working Properly |
| TST-007 | Security Test | All API calls should be secure and use HTTPS | All API calls are secure and use HTTPS | Passed | Working Properly |
| TST-008 | Image Loading Test | All car images should load correctly | All car images loaded without issues | Passed | Working Properly |
| TST-009 | Accessibility Test | Site should meet accessibility standards | Site meets all accessibility standards | Passed | Working Properly |
| TST-010 | Responsiveness Test | Site should adjust properly on various devices | Site adjusted properly on all tested devices | Passed | Displayed Properly |