Vercel Test Cases	Pre condition	Expected Resul	Actual Result	Post condition	Pass/Fail	Test Owner
Test the functionality of the environment variables defined in the Vercel platform.						
Test the proper loading of static assets such as images and stylesheets.						
Test the proper behavior of API endpoints, including correct response codes and payloads.						
Test the handling of incoming webhooks and ensure that they are processed correctly.						
Test the proper behavior of user authentication, including login and logout.						
Test the performance of the application under different traffic loads and ensure it remains responsive.						
Test the compatibility of the application with different browsers and devices.						
Test the handling of server-side errors and ensure that they are logged and reported correctly.						
Test the behavior of the application when network conditions change, such as slow or unreliable connections.						
Test the proper integration with other services and APIs, such as databases and email.						
Test the integration of the application with the Vercel platform, including deployment, scaling, and resource allocation.						
Test the integration of the application with other services and APIs, such as databases, email providers, and payment gateways.						
Test the integration of the front-end and back-end components, including the behavior of the user interface and the exchange of data between the two.						
Test the end-to-end flow of the application, including user sign-up, login, and account management.						
Test the integration of the application with third-party tools and services, such as analytics and marketing platforms.						
Test the integration of security features, such as SSL certificates and encryption, to ensure sensitive information is protected.						
Test the integration of the application with other Vercel services, such as Functions and Serverless APIs.						
Test the behavior of the application under different network conditions, such as slow or unreliable connections.						
Test the integration of the application with other deployment environments, such as local development and staging.						
Test the integration of the application with monitoring and logging services, such as error reporting and performance metrics.						
Test the implementation of authentication and authorization mechanisms to ensure only authorized users can access sensitive data.						
Test the handling of sensitive information, such as passwords and credit card numbers, to ensure it is encrypted and stored securely.						
Test the application for vulnerabilities such as cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection.						
Test the proper configuration of security-related HTTP headers, such as X-XSS-Protection, X-Content-Type-Options, and X-Frame-Options.						
Test the application's response to malicious requests and payloads to ensure it does not disclose sensitive information or allow unauthorized access.						
Test the integration of the application with SSL/TLS certificates to ensure secure communication between the server and client.						
Test the application for vulnerabilities related to session management, such as session hijacking and fixation.						
Test the application's response to security incidents, such as a server breach or unauthorized access, and ensure it implements proper recovery procedures.						
Test the proper configuration of firewalls and network security to ensure the application is protected against attacks from the outside.						
Test the proper implementation of data backup and recovery procedures to ensure sensitive information can be recovered in case of an outage or disaster.						
Test the application's response time and load times under normal traffic conditions to ensure it remains responsive for users.						
Test the scalability of the application, including the ability to handle increasing traffic and resource demands.						
Test the performance of the application under heavy load conditions, such as during high traffic periods or when processing large amounts of data.						
Test the performance of the application with different configurations, such as varying numbers of server instances and resources.						
Test the performance of the application when running in different environments, such as local development and production.						
Test the performance of the application when integrating with other services and APIs, such as databases and payment gateways.						
Test the performance of the application under different network conditions, such as slow or unreliable connections.						
Test the performance of the application when processing large amounts of data, such as uploading and downloading large files.						
Test the performance of the application when processing high volumes of requests, such as during high traffic periods.						
Test the performance of the application when implementing new features and updates to ensure they do not negatively impact performance.						
Test the full flow of the application from start to finish, including user sign-up, login, and account management.						
Test the integration of the application with other services and APIs, such as databases, email providers, and payment gateways.						
Test the behavior of the application under different network conditions, such as slow or unreliable connections.						
Test the compatibility of the application with different browsers and devices, including mobile and desktop.						
Test the behavior of the application when processing large amounts of data, such as uploading and downloading large files.						
Test the behavior of the application when processing high volumes of requests, such as during high traffic periods.						
Test the end-to-end security of the application, including proper implementation of authentication and encryption.						
Test the behavior of the application when implementing new features and updates to ensure they do not negatively impact existing functionality.						

Test the end-to-end performance of the application, including response times, load times, and scalability.						
Test the proper handling of server-side errors and ensure that they are logged and reported correctly.						
Test the application's compliance with accessibility guidelines, such as the Web Content Accessibility Guidelines (WCAG) and the Accessible Rich Internet Applications (ARIA) specification.						
Test the application's behavior with screen readers and other assistive technologies to ensure it is usable for users with disabilities.						
Test the color contrast and font size of text and other elements in the application to ensure they are accessible to users with visual impairments.						
Test the behavior of the application with keyboard-only navigation to ensure it is usable for users who cannot use a mouse.						
Test the labeling and descriptions of form fields and other controls to ensure they are accessible to users with cognitive and motor impairments.						
Test the behavior of the application when using alternative input methods, such as voice commands and switch controls.						
Test the behavior of the application when using alternative output methods, such as audio descriptions and braille displays.						
Test the behavior of the application with different browsers and devices, including mobile and desktop, to ensure it is accessible on a variety of platforms.						
Test the application's behavior with users who have low vision or color blindness, including the use of clear and distinct colors and patterns.						
Test the application's behavior when using accessibility features, such as high contrast modes, to ensure they work correctly.						