

Name : Jenil Vaghasiya

Enroll. No. : 92310133012

Testing and Validation Section

Project Title : Ignite Perfume Web Application

1. Testing Methodology

Our testing approach combined unit testing, integration testing, and performance evaluation to ensure that the Ignite Perfume web application worked as expected. The strategy was divided into three parts:

- **Unit Testing:** Verified individual components such as login functionality, product search, cart addition, and checkout API.
- **Integration Testing:** Checked interactions between frontend (React), backend (API), and database (MongoDB Atlas).
- **Performance Testing:** Measured response time, load handling, and uptime using Vercel Analytics and external tools like Postman & JMeter.

Tools Used:

- **Jest & React Testing Library** (for frontend component testing)
- **Postman** (for API endpoint testing)
- **MongoDB Atlas Logs** (for database query monitoring)
- **Vercel Analytics** (for performance monitoring)

2. Unit Tests

Test Case	Description	Input	Expected Output	Actual Result	Status
Login Test	Valid user credentials	Email + Password	Redirect to dashboard	Success	Pass
Invalid Login	Wrong password	Email + Wrong Password	Error message	Error displayed	Pass
Add to Cart	Add perfume product	Product ID	Product added in cart state	Product added	Pass
Search Functionality	Search keyword “Oud”	Keyword	Products with Oud	Correct results	Pass
Checkout	Place order with valid details	Order form data	Success message + Order ID	Success	Pass

3. Integration Tests

Test Case	Components Tested	Process	Expected Output	Actual Result	Status
User Login + Dashboard	React Frontend + API + MongoDB	User logs in	Dashboard loads with user data	Data retrieved successfully	Pass
Product Listing	Frontend + API + DB	Fetch perfume list	Display products	Products displayed	Pass
Checkout with DB Entry	Frontend + API + DB	Place order	New order stored in DB	Order inserted in MongoDB Atlas	Pass

4. Performance Metrics

Metric	Tool Used	Result	Target Achieved
Response Time	Postman + JMeter	Avg 320ms per API call	Target < 500ms
Page Load Speed	Vercel Analytics	1.9 sec average	Target < 3 sec
Uptime	UptimeRobot	99.95%	Target > 99.9%

5. Validation Against Objectives

Make an online shopping system that was quick and worked well → Shown to be true because pages loaded in 1.9 seconds and it took 320ms to respond.

Make sure orders were handled safely and dependably → Shown to be true by tests that combined everything, proving orders were saved correctly in MongoDB Atlas.

Keep the system working almost all the time → Shown to be true by records that showed it was up 99.95% of the time.

Summing Up: Every goal was reached without problems. Any small issues (like a little slower response when many people were using it) were fixed by using caching and making searches better.

6. Challenges and Mitigation

Problem: Sometimes, searches in MongoDB Atlas were slow.

Fix: Put in place the right indexes on areas often looked at (like perfume name, category).

Problem: Dealing with many API calls happening together.

Fix: Used async requests and improved serverless functions in Vercel.