

# PoC Hypotheses Report for IdealShop Transactional Web Application

**Course:** TRANSACTIONAL WEB APPLICATIONS

**Project:** IdealShop — E-Commerce Website with Admin and Customer Panels

**Author:** Reza Akbari

---

## Introduction

This Proof of Concept (PoC) report aims to test and evaluate three hypotheses regarding the IdealShop e-commerce system. IdealShop is a transactional web application built using ASP.NET Core, Entity Framework, and SQL Server. The system supports user roles, product management, cart functionality, secure login/logout features, and an order workflow.

The selected hypotheses for testing are related to system **performance**, **usability**, and **deployment**. These areas are essential to ensure the system's technical feasibility and its ability to meet real-world transactional needs.

---

## Hypothesis 1: Performance

**Statement:** "An ASP.NET Core web application using SQL Server can process customer checkout requests at least 15% faster than a Node.js application using MongoDB under identical transactional load."

### Purpose

To validate whether the chosen stack (ASP.NET Core + SQL Server) is efficient in handling transactional requests such as cart checkout.

### PoC Approach

- Build two comparable web applications: one using ASP.NET Core and SQL Server, and the other using Node.js and MongoDB.
- Simulate 100 concurrent checkout requests using a performance testing tool like Apache JMeter or Postman.
- Measure and compare the average response times of both systems.

### Expected Result

IdealShop (ASP.NET Core) should demonstrate at least 20% faster processing under load due to its optimized server-side rendering and efficient ADO.NET-based data operations.

---

## Hypothesis 2: Usability

**Statement:** "The Razor Pages interface provides a smoother and more intuitive user experience for admins managing product inventory, as measured by a 25% lower task completion time compared to a React-based UI."

## Purpose

To determine whether the current Razor Pages UI is effective and easy to use for the admin user when managing data.

## PoC Approach

- Define three typical tasks: Add Product, Edit Product, and Delete Product.
- Prepare two UI versions: one with Razor Pages, one with React.
- Recruit 5 test users to complete these tasks on both systems.
- Record task completion time and satisfaction feedback.

## Expected Result

The Razor Pages UI is expected to yield faster and more efficient task completion due to tight integration with the backend and form validation.

---

## Hypothesis 3: Deployment & Maintenance

**Statement:** "Dockerized deployment using Azure App Service requires 30% less setup and configuration time than deploying manually to AWS EC2."

## Purpose

To evaluate the ease and speed of modern container-based deployment methods versus traditional virtual machine-based deployment.

## PoC Approach

- Deploy the application on Azure App Service using Docker containers.
- Deploy the same application manually on AWS EC2.
- Measure total time taken including VM setup, environment configuration, and successful deployment.

## Expected Result

The containerized Azure deployment should reduce setup time significantly due to preconfigured build pipelines and environment abstraction.

---

## Conclusion

These three hypotheses aim to validate IdealShop's architecture choices in terms of performance, user experience, and ease of deployment. Results from the PoC tests will guide the decision-making process regarding technology stack, interface design, and hosting strategies.

Further testing and analysis will be documented in the upcoming project phases, including detailed metrics, charts, and user feedback from the PoC exercises.