iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.82832

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Secure API Gateway Implementation with JWT for E-Commerce

Nujat Najir Mulani[1], Dr. Namrata Ghadgil[2]
[1]*DepartmentofMasterofComputerApplication, JSPM University, Pune, Maharashtra, India*

***Abstract:** In this study, we analyze the designing and development process of a safe e-commerce platform built on microservices architecture. As e-commerce platforms becomeincreasinglyimportant,thereisanevidentneedfor webapps whichshould beeffective, safe,andeasytoscale. At the same time, monolith applications present some obvious disadvantages due to increasing demand, such as complicated maintenance, difficulty in scaling, and security challenges. In order to cope with these problems, more and more people tend to implement microservices.*
Firstly, the paper incorporates a number of new technologies within one architecture. Firstly, an API Gateway and JWT authentication ensure strict and solid security. The frontend is realized through React which ensures high-level interactivity. Moreover, Java Spring Boot works on the backend allowing to separate different services whereas Spring Cloud Gateway acts as a main gatewayresponsibleforroutingandsecurityamongclients and services. Besides, data storage is provided by MongoDBtoensureflexibilityandscalability.Inthisway, the protection of APIs is performed through JSON Web Token (JWT) authentication ensuring users' credentials. Finally, to deploy this application, Docker is used to controlcontainers,ensuringtheworkandcommunication among services.
Ingeneral,thisstudyallowsustounderstandhowtocreate a safe and scalable e-commerce platform with modern technologies.
Index Terms—API Gateway, JWT, Microservices, Spring Boot, React, MongoDB, Docker, E-Commerce Security.

I. INTRODUCTION

Modern e-commerce solutions represent an essential component of the operation of any business organization. Customers demand fast load speed, security,andhighavailabilityofthesystem.Thistype of software processes very sensitive data, such as personallogindetails,customerinformation,payment histories, etc., which means that the main concern associated with building this kind of web application is providing a strong protection mechanism.
Most e-commerce applications have a monolithic architecturethatintegratesallmajorcomponentssuch as users' administration, product catalogue, payment processing, and order processing into one big application. Although this model can work well for small scale applications, it does not allow scaling or modifying the app in future.
However, moving to microservice architecture changes the situation. The software is divided into services responsible forparticular tasks. Thisapproach allows better scalability and easier development but also involvesthe problemof inter-service interaction.
A central entrance point for client requests is API Gateway that handles routing, authorization, logging, and monitoring. In this case, we integrate our API Gateway with JWT to verify a user before accessing other services.
Theaimofthepresentresearchistodevelopasecure andscalablesolutionforane-commercesitethatuses React for the frontend, Spring Boot microservices on thebackend,SpringCloudGateway,JWT,MongoDB, and Docker.

II. LITERATURE REVIEW

Numerous studies emphasize the significance of security at the level of the Application Programming Interface (API). With microservices architecture, the useofAPIGatewaysiswidelyspreadastheysimplify communication between the front- and back-end servers.
Anotherinterestingtopic,thatshouldbementioned,is JWT authentication. This method of identification of the user is widely used for REST APIs due to its simplicity compared to session-based authentication, as well as absence of necessity in storage of session information on the side of the server.
Speaking of securing enterprise applications, such librariesasSpringSecurityandSpringCloudGateway are commonly used as they provide not only identification via tokens but also route management, filtering of requests, and authorization of the user.
Finally, Docker and other container tools revolutionized application development, as now the developer is guaranteed to have no problems of transferring an application from testing to production environment.

Considering the abovementioned facts, one can suggest that API Gateway, JWT, microservices, and containerization will be a good choice in case of e-commerce development. Such combination has been proven effective by numerous studies.

III. PROBLEM STATEMENT

Thetypicalweb-basedbusinessapplicationfacesa number of problems simultaneously, such as:

- Challengesinscalingwhenfacinghighlevelsof traffic
- SecurityrisksthatcomewithusingAPIsdirectly
- Challengesinmaintenanceduetotightmodule coupling
- Slowdeploymentprocess
- Limitedflexibilityforfuturechanges

As a result, it is necessary to design a flexible, secure systemwhichwillbeabletohandletrafficeffectively while keeping APIs and user data safe.

IV. OBJECTIVES

Objectivesoftheprojectincludethefollowing:

- TodevelopanAPIGatewaywithsecurity capabilities for an e-commerce application.
- Tocreateindependentmicroservicesbyusing Spring Boot.
- Tointegrateauthenticationandauthorisationby using JSON Web Tokens (JWT).
- To developadynamic userinterface by employing React.
- TouseMongoDBtostoretheapplication'sdata.
- Topackagetheapplication'sservicesusingDocker containers.
- To improve security, scalability, and maintainability.

V. SYSTEM ARCHITECTURE

This project consists of four levels of hierarchy, each having different functional responsibilities.

A. Front End Level

This level is designed using React and includes user interface for:

- Home Page
- ProductViewing
- CartManagement
- Registration&Login
- OrderHistory
- AdminDashboard

B. APIGatewayLevel

SpringCloudGatewayactsastheentrypointfor handling all requests and is responsible for:

- RequestRouting
- TokenValidation
- Filtering
- Security
- LoadBalancing

C. Micro-servicesLevel

The back end layer consists of multiple independent micro-services such as:

- UserService:Thisservicetakescareofuserregistration, authentication and their data
- ProductService:Thisservicecontrolstheproduct details and inventory
- OrderService:Handlesallthecustomerorder processing
- PaymentService:Responsibleforpaymentprocessing

D. Data Layer

MongoDBdatabasecontainsthedatadomains of:

- UserDomain
- ProductDomain
- OrdersDomain
- PaymentDomain

E. ExecutionLayer

All the microservices are packed and deployed as Docker containers.

VI. METHODOLOGY

Step1:Front-endDevelopment

React framework is used to develop a dynamic front end that allows users to view their products, additems to the cart, place orders, and manage account details.

Step2:Back-endDevelopment

A different Spring Boot microservice is developedfor each business function.

Step3:Authentication

Users log in using their usernames and passwords to receive JSON Web Tokens on successful authentication.

Step4:APISecurity

The client sends a JSON web token in the Authorization header in all secure requests, which is verified by the API Gateway before sending arequest.

Step5:DatabaseManagement

Data is stored in MongoDB, which uses JSON-like documents, thus making e-commerce possible.

Step6:Containers

Containers are created for all components using Docker to enable hassle-free deployment.

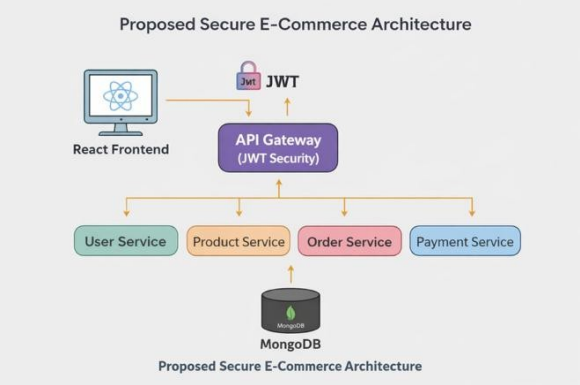


Fig1.ProposedSecureE-CommerceArchitecture

VII. IMPLEMENTATION

A. Technologies Used

Components	Technology
Frontend	React.js
Backend	SpringBoot
Security	JWT
Gateway	SpringCloudGateway
Database	MongoDB
Containerization	Docker
APITesting	Postman

B. SecurityProcessFlow

Step1:Usercredentialsubmission
Step2:Authenticationofusercredentialsubmission
Step3:GenerationofJWTtokenincaseofsuccessful authentication
Step4:Storageofthetokenattheclient-side
Step5:Inclusionoftokenbytheclientinall subsequent requests
Step6:Validationoftokenbythegateway
Step7:Requestroutingtocorrespondingservices

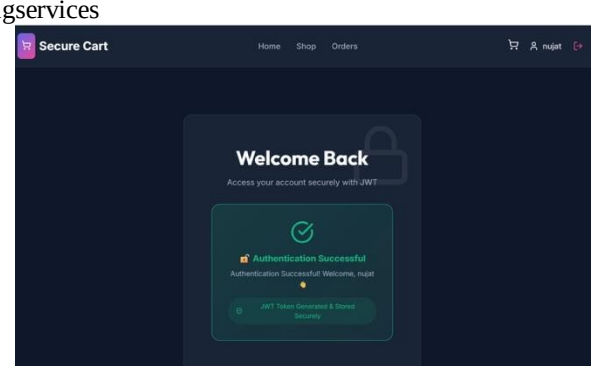


Fig2.UserLoginInterface

VIII. RESULT AND OUTPUT

Capabilitiesofthesysteminclude:

- Userauthentication
- Role-basedaccessforadminsandusers
- Abilitytolistandsearchforproducts
- Shoppingcartmanagement
- Orderingprocessmanagement
- Securityfromanyformofunauthorizedaccess
- ProtectionoftheAPIusingJSONwebtokens(JWT)
- Aresponsivefrontend
- EfficientroutingthroughtheuseofanAPIGateway

Moreover,theadminpanelallowsforthemanagement of products and user rights, whereas the customer-facing platform enables safe shopping and order management.

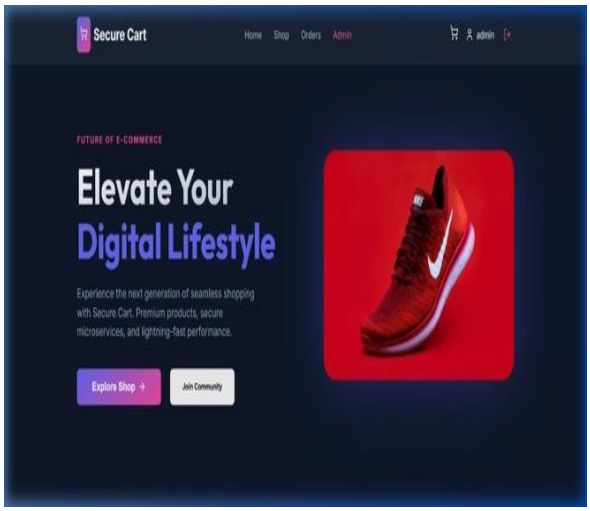


Fig3.HomePage

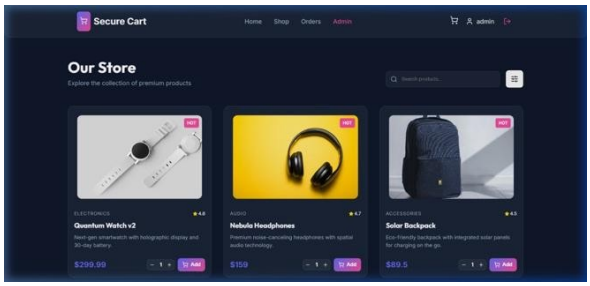


Fig4.ProductListingModule

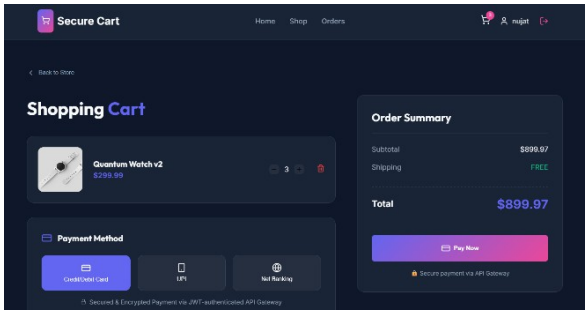


Fig5.Cart&Checkout

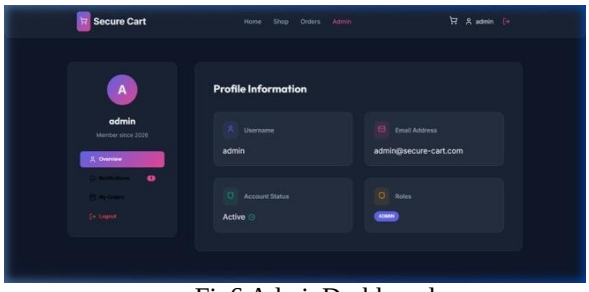


Fig6.AdminDashboard

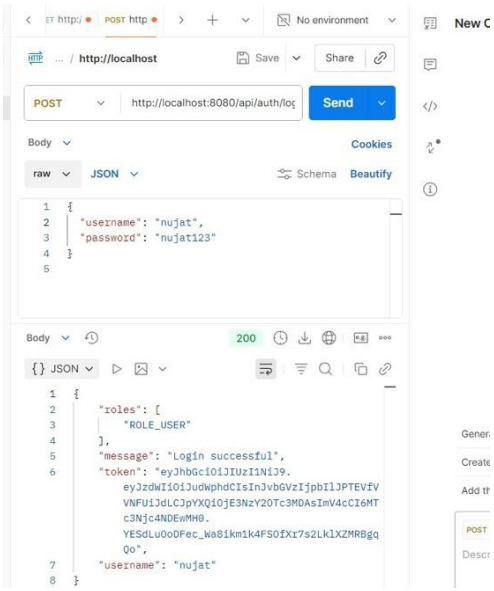


Fig7.APITestingwithJWTToken

IX. ADVANTAGES OF PROPOSED SYSTEM

- HighsecuritythroughJWTauthentication
- CentralizedAPImanagement
- Easyscalabilityusingmicroservices
- Independentservicedeployment
- Betterfaultisolation
- Fasterdevelopmentandupdates
- Modernresponsiveuserexperience
- SimplifieddeploymentwithDocker

X. FUTURE SCOPE

Thesystemcanbeenhancedfurtherbyadding:

- Paymentgatewayintegration
- RecommendationengineusingAI
- Real-timenotifications
- Kubernetesdeployment
- Multi-vendormarketplacesupport
- Analyticsdashboard
- OAuth2andsociallogin
- Advancedmonitoringtools

XI. CONCLUSION

In this research, the success implementation of an e-commerceapplicationhasbeenillustratedthatutilizes the latest web technologies in terms of security and other performance attributes. The use of ReactJS frontend, microservices in Spring Boot, API gateway in Spring Cloud Gateway, MongoDB, JSON Web Token for security of APIs, and Docker make the platform very secure, scalable, and flexible as well.

The API Gateway provides a single place for route management and authentication purposes, whereas JSONWebTokensmakesurethataccesstoprotected APIs is secured.

XII. ACKNOWLEDGMENT

I want to say thank you to everyone who helped with the research project called Implementation of Secure API Gateway using JWT Authentication for an E-commerce Portal.

I am really thankful to Dr. Namrata Gadgil because she guided me and gave me suggestions at everystage of the Implementation of Secure API Gateway using JWT Authentication for an E- Portal project.

She helped me a lot with the part and the documentation part of the Implementation of Secure API Gateway using JWT Authentication for an E-commerce Portal project.

I also want to thank the teachers at the department of computer applications at JSPM university in Pune.

They helped me a lot, with my studies. Gave me the support and facilities I needed to do the Implementation of Secure API Gateway using JWT Authentication for an E-commerce Portal project.

Myfriendsandcolleagueswerealsoveryhelpful.

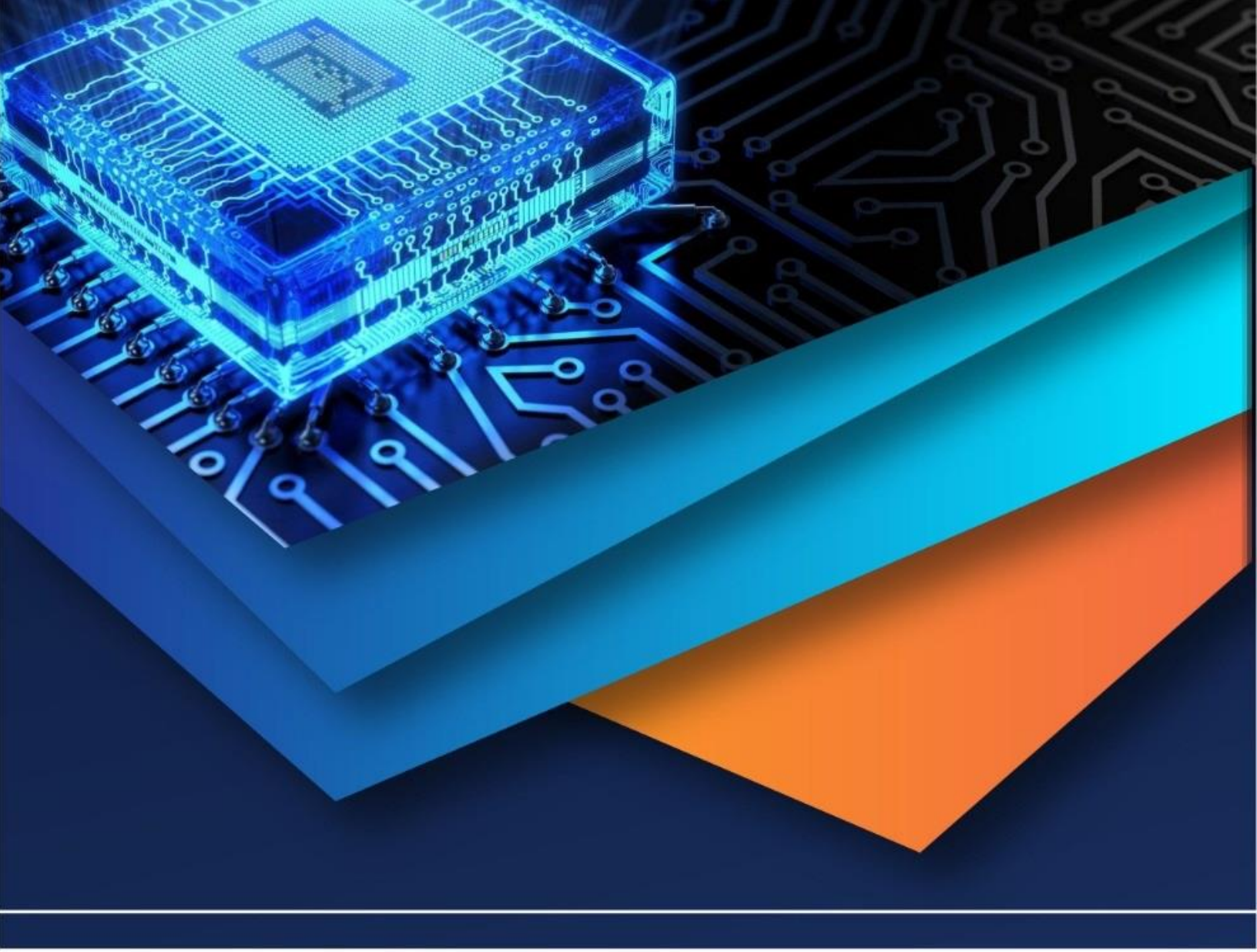
They gave me suggestions when we talked about the Implementation of Secure API Gateway using JWT AuthenticationforanE-commercePortalproject and that helped me a lot.

I also want to thank my family for encouraging me and motivating me to do the Implementation of Secure API Gateway using JWT Authentication for an E- Portal project.

REFERENCES

[1] S. Newman, Building Microservices: Designing Fine-Grained Systems. Sebastopol, CA, USA: O'Reilly Media, 2021.
[2] C. Richardson, Microservices Patterns: With ExamplesinJava.ShelterIsland,NY,USA:Manning Publications, 2018.
[3] Spring Team, "Spring Boot Documentation," VMware, Inc. [Online]. Available: https://spring.io/projects/spring-boot

[4] Spring Team, "Spring Cloud Gateway Documentation," VMware, Inc. [Online]. Available: https://spring.io/projects/spring-cloud-gateway
[5] Auth0, "Introduction to JSON Web Tokens (JWT)." [Online]. Available: https://jwt.io/introduction
[6] MongoDB Inc., "MongoDB Official Documentation." [Online]. Available: https://www.mongodb.com/docs/
[7] DockerInc.,"DockerDocumentation."[Online]. Available:https://docs.docker.com/
[8] Meta Platforms Inc., "React Official Documentation." [Online]. Available: https://react.dev/
[9] O.TilkovandS.Vinoski,"Node.js:Using JavaScripttoBuildHigh-PerformanceNetwork Programs,"IEEEInternetComputing,vol.14,no.6, pp.80–83,Nov.-Dec.2010.
[10] M. Fowler and J. Lewis, "Microservices: A Definition of This New Architectural Term," Martin Fowler, 2014. [Online]. Available: https://martinfowler.com/articles/microservices.html



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY