Integrating Web Development and RPA: Automating Data Handling for E-commerce Platforms

Harshith Gowda HP[1], Sumanth[2], Lokesh[3], Dr. Nihar Ranjan Nayak[4]
[1,2,3,4]*Presidency School of Computer Science and Information Science Presidency University, Itgalpura, Rajanukunte, Bengaluru-560064*

Abstract—**This paper introduces the design and implementa- tion of an e-commerce platform that integrates web development technologies with robotic process automation (RPA), aiming to enhance user experience and streamline backend processes. The proposed architecture leverages a responsive Angular front-end, a Spring Boot back-end, and a MySQL database for data management, alongside UiPath workflows to automate repetitive tasks. The front-end enables users to browse products and place orders, while the back-end secures data handling and communication through REST APIs. Meanwhile, UiPath work- flows interface directly with the database, automating operations such as retrieving user and order information and generating automated email alerts.**

The demonstrated system architecture showcases modularity, scalability, and efficient data handling. Empirical findings in- dicate that incorporating UiPath automation reduces manual tasks and ensures timely, accurate notifications, contributing to a more efficient workflow. The research underscores the viability of combining contemporary web frameworks with RPA solutions for addressing practical challenges in e-commerce. Future extensions may involve real-time data updates and AI- driven recommendations to further elevate system performance and user satisfaction.

Index Terms—**Web Development, Angular, Spring Boot, REST API, MySQL, UiPath, Robotic Process Automation, E-commerce, Automation, Workflow, Data Handling**

I. INTRODUCTION

The rapid rise of web development and automation tech- nologies has profoundly reshaped digital ecosystems, espe- cially within the domain of e-commerce. Modern web frame- works like Angular and Spring Boot simplify the creation of dynamic interfaces, improving user interactions and satisfaction. Angular, a front-end framework widely employed in building single-page applications (SPAs), streamlines the development of rich, responsive user experiences. Spring Boot, on the other hand, enables developers to build robust back- end services swiftly, complete with extensive tooling for scalability.

Concurrently, Robotic Process Automation (RPA) has emerged as a core technology for automating tasks that are repetitive and rules-based, thereby minimizing manual effort and mitigating errors. By simulating human-computer inter- actions, RPA solutions can operate across diverse software ecosystems. Research has shown that RPA accelerates digital transformation and enhances productivity in numerous indus- tries [1] [2].

The integration of web frameworks with RPA offers a promising strategy for designing e-commerce platforms that deliver both user-friendly interfaces and operational efficiency. Leveraging Angular and Spring Boot for front-end and back- end development, along with an RPA tool such as UiPath, organizations can build solutions that automate crucial pro- cesses like order handling, data entry, and notifications. These approaches have been demonstrated to lower operational costs and optimize system performance. Furthermore, a relational database management system like MySQL supports reliable data storage, query processing, and maintainability.

This paper explores the synergy between Angular, Spring Boot, MySQL, and UiPath to create an automated e-commerce platform. It delves into the system’s underlying architec- ture, implementation details, and the resultant advancements in workflow efficiency and user contentment. By doing so, it highlights the opportunities and complexities involved in merging web development practices with

RPA in the dynamic e-commerce environment.

II. RELATED WORK

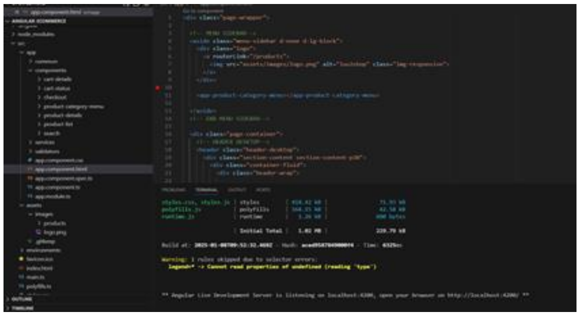
The Angular framework is widely employed in developing SPAs due to its robust client-server communication features. This approach enables seamless data interchange, significantly improving web application performance and user experience [3]. Research underscores Angular’s modular architecture, which fosters maintainable codebases [4]. Nevertheless, de- signing and documenting REST APIs demands thorough consideration of performance and security, as highlighted in studies on REST specifications [5].

In parallel, RPA has become a cornerstone in automating manual, repetitive tasks, leading to gains in efficiency and reductions in error. UiPath, one of the leading RPA plat- forms, specializes in optimizing business processes through user-friendly interfaces and robust automation functionalities. Integrating RPA and AI solutions is frequently cited as a key facet of Industry 4.0, offering new pathways for productivity and innovation [6]. While RPA tools like UiPath effectively automate repetitive tasks, seamless integration with existing web architectures can be challenging, especially regarding data extraction and manipulation. Researchers emphasize the need for a well-defined strategy when embedding RPA functionali- ties into web-based systems [7] [8].

Despite ample insights into web frameworks and RPA software, fewer studies focus on marrying these disciplines to automate back-end processes. This project bridges this gap by demonstrating an end-to-end solution that combines Angular for the front-end, Spring Boot for the back-end, RESTful APIs for client-server communication, and UiPath for automating back-end operations such as data retrieval and email notifications. This unification aims to reduce manual tasks, facilitate consistent data handling, and deliver prompt user notifications.

III. METHODOLOGY

A. Front-End (Angular)



```
Warning: 1 rules skipped due to selector errors:
  legend+* -> Cannot read properties of undefined (reading 'type')

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
```

Fig. 1. Angular code layout in VSCode.

1. Angular Framework Setup and Design: Angular pro- vides a component-based structure for building SPAs. Project initialization utilizes Angular CLI, followed by module con- figuration and component generation.

2. UI/UX Styling: Bootstrap is integrated for responsive layout and pre-styled components, while Font Awesome sup- plies scalable icons.

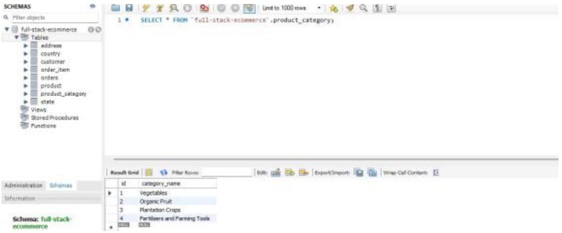
B. Back-End (Spring Boot)

Fig. 2. Spring Boot project structure in IntelliJ, showing controllers, reposi- tories, and entities.

1. Spring Boot REST APIs: Spring Boot establishes the core business logic and routes for handling HTTP requests. The REST APIs incorporate services for user authentication, product management, and order processing.

A. Database (MySQL)

Entities for users, products, and orders are structured in MySQL to ensure data consistency and swift query responses.



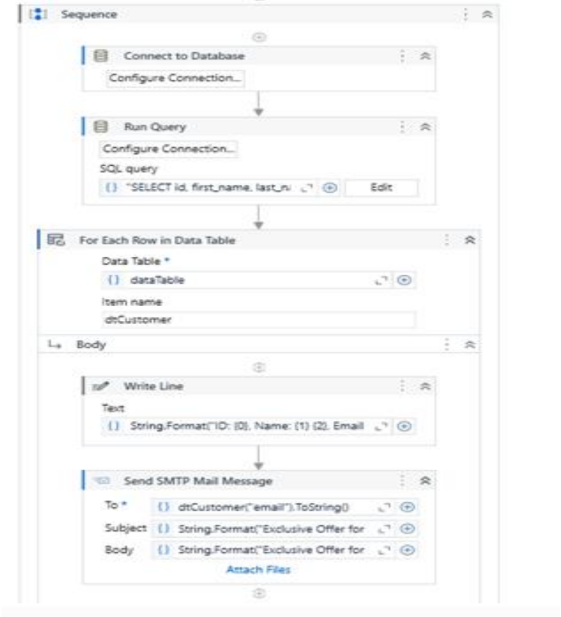
```
SELECT * FROM `full-stack-ecommerce`.product_category;
```

id	category_name
1	Vegetables
2	Organic Fruit
3	Plantation Crops
4	Fertilizers and Farming Tools
NULL	NULL

Schema: full-stack-ecommerce

Fig. 3. MySQL database schema (full-stack-ecommerce) with product, order, and customer tables.

B. RPA Workflow (UiPath)



Sequence

Connect to Database
Configure Connection...

Run Query
Configure Connection...
SQL query
"SELECT id, first_name, last_n
Edit

For Each Row in Data Table
Data Table *
dataTable
Item name
dtCustomer

Body

Write Line
Text
String.Format("ID: {0}, Name: {1} {2}, Email

Send SMTP Mail Message
To * dtCustomer("email").ToString()
Subject String.Format("Exclusive Offer for
Body String.Format("Exclusive Offer for
Attach Files

Fig. 4. UiPath workflow querying customers from the database and sending automated emails.

1. Database Connection: UiPath directly connects to the MySQL database, running SQL queries to retrieve and update data automatically.

2. Email Workflow: Automated email alerts are triggered in the Main.xaml workflow, notifying customers of order updates and confirmations without human intervention.

C. Integration: Final UI and Customer Experience

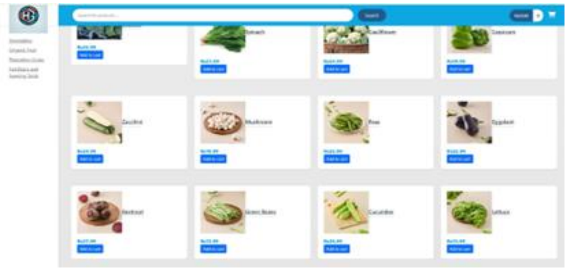


Fig. 5. Integrated e-commerce website UI, combining Angular (front-end), Spring Boot (back-end), and MySQL (data).

1. Angular Front-end dispatches requests via REST APIs to the Spring Boot back-end.

2. Spring Boot Back-end interprets the requests and com- municates with the MySQL database for data retrieval or insertion.

3. UiPath continuously monitors the database, sending out automated emails based on pre-defined triggers.

IV. SYSTEM ARCHITECTURE

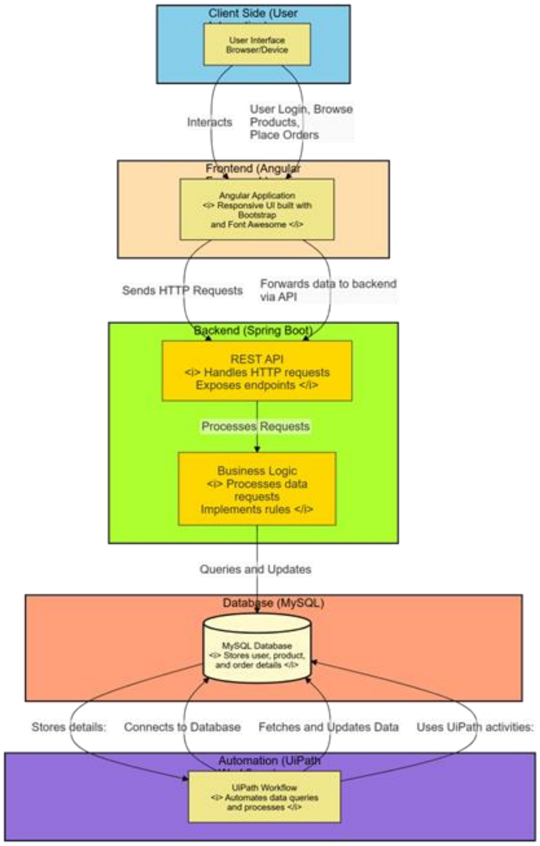

Client Side (User
User Interface Browser/Device
Interacts
User Login, Browse Products, Place Orders
Frontend (Angular
Angular Application <i> Responsive UI built with Bootstrap and Font Awesome </i>
Sends HTTP Requests
Forwards data to backend via API
Backend (Spring Boot)
REST API <i> Handles HTTP requests Exposes endpoints </i>
Processes Requests
Business Logic <i> Processes data requests Implements rules </i>
Queries and Updates
Database (MySQL)
MySQL Database <i> Stores user, product, and order details </i>
Stores details:
Connects to Database
Fetches and Updates Data
Uses UiPath activities:
Automation (UiPath
UiPath Workflow <i> Automates data queries and processes </i>

Fig. 6. Overall system architecture integrating Angular, Spring Boot, MySQL, and UiPath

The system comprises four main layers:

1. Client Side: End-users interact through a browser or similar interface. User actions (browsing, ordering, ac- count management) are propagated to the front-end.

2. Front-end (Angular): Implements the user interface with Angular, employing REST calls to the back-end. Styling is handled by Bootstrap, and icons come from Font Awesome.

3. Back-end (Spring Boot): Executes core business logic and exposes REST APIs for data operations.

4. Database (MySQL): Stores all essential data, ensuring reliable querying and persistence.

5. Automation Layer (UiPath): Operates as a specialized automation solution, directly connecting

to the MySQL database to manage repetitive tasks and dispatch email notifications.

V. RESULTS AND DISCUSSION

A. Results

The finalized system effectively integrates a dynamic e-commerce application with automated workflows. The Angular-based interface ensures a smooth user experience, while Spring Boot reliably manages data exchange and en- forces security. MySQL upholds performance and integrity in data transactions. UiPath automation markedly reduces manual intervention in tasks like order confirmations, thereby improving precision and timeliness.

B. Discussion

This work demonstrates that coupling RPA with web frame- works can substantially reduce operational overhead in e- commerce contexts. By delegating routine tasks to UiPath, businesses can pivot to more strategic activities. However, adopting an RPA layer may require careful planning for data synchronization, scaling, and security concerns.

VI. CONCLUSION

This paper underscores the advantages of merging web development frameworks (Angular and Spring Boot) with RPA technologies (UiPath) to construct a scalable, efficient e-commerce platform. Results reveal gains in performance, accuracy, and user engagement, attributable to automated processes and seamless client-server communication. Future endeavors could introduce real-time updates via WebSockets, intelligent product recommendations powered by AI, and more extensive automation workflows. In essence, this integration model lays the groundwork for sophisticated, next-generation e-commerce solutions.

REFERENCES

[1] J. Siderska, “Robotic Process Automation—A driver of digital transfor- mation?” Engineering Management in Production and Services, vol. 12, no. 2, pp. 21–31, 2020.

[2] P. Hofmann, C. Samp, and N. Urbach, “Robotic process automation,”

[3] Electronic Markets, vol. 30, pp. 99–106, 2020.

[4] H. H. Ngoc, “Single Page Web Application with Restful API and AngularJS: Best Practices with Verto Monitor,” Helsinki Metropolia University of Applied Sciences, 2014.

[5] J. Cincovic, S. Delcev, and D. Draskovic, “Architecture of Web Ap- plications Based on Angular Framework: A Case Study,” University of Belgrade, School of Electrical Engineering, 2019.

[6] J. Foster and J. Foster, “A Qualitative Study of REST API Design and Specification Practices,” Tufts CS, 2023.

[7] J. Ribeiro, R. Lima, T. Eckhardt, and S. Paiva, “Robotic Process Automa- tion and Artificial Intelligence in Industry 4.0 – A Literature Review,” Procedia Computer Science, vol. 181, pp. 51–58, 2021.

[8] H. Saini and P. Kaur, “A Comparative Analysis of RPA Tools: UiPath, Automation Anywhere, and Robocorp,” in IEEE Conference Proceedings, 2022.

[9] Y. Bhagwat and A. Sharma, “Data Capturing in Web Applications Using RPA UiPath,” International Journal of Research Publication and Reviews, vol. 4, no. 6, pp. 1456–1461, 2021.