"COMPKILLER"

Comprehensive Real-Time E-commerce Price Tracker

SYNOPSIS OF MINI PROJECT
of
Bachelor of Technology
in
Computer Science & Engineering
By

S. NO	Student Name	Roll Number	Branch and Section
1	**Sourav Singh**	2001650100087`	CS-B

Kanpur Institute of Technology, Kanpur
Dr. A.P.J Abdul Kalam Technical University, Lucknow

<u>DECLARATION</u>

I hereby declare that this submission of my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma or the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature :

Name: Sourav Singh

Roll. No : 2001650100087

Date :

<u>**Submitted To:**</u>

Mr. Balram Sir
Assistant Professor
Kanpur Institute of Technology, Kanpur

<u>ACKNOWLEDGEMENT</u>

It gives me a great sense of pleasure to present the report of the B.Tech mini project done during B.Tech Final Year. I owe a special debt of gratitude to Mr. Balram Sharma for his constant support and guidance throughout the course of my work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only their cognisant efforts that my endeavour's have been light of the day. I also take the opportunity to acknowledge the contribution of Mr. Rahul Singh , Head of Department of Computer Science & Engineering, Kanpur Institute of Technology, Kanpur (U.P.) for his full support and assistance during the development of the mini project. I would also like to thank the contribution of all faculty members of the department for their assistance and cooperation during the development of my mini project.

Signature:
Name: Sourav Singh
Roll no: 2001650100087
Date:

ABSTRACT

The "COMPKILLER" project introduces a sophisticated real-time e-commerce price tracking tool designed to revolutionise the way users monitor product prices on Amazon. With a focus on dynamic and advanced features, COMPKILLER enables users to stay informed about the ever-changing landscape of product pricing. By implementing a real-time web scraper using cron techniques, the tool gathers data at regular intervals, creating a comprehensive price history that includes current, average, lowest, and highest prices. This functionality empowers users to make well-informed purchasing decisions by providing a holistic view of pricing trends and enabling strategic choices based on accurate and up-to-date information.

Utilising Next.js, Node.js, and Express.js, COMPKILLER boasts a robust technological foundation for its implementation. The system architecture ensures seamless integration of features, and the integration of external services, including nodemailer, cheerio, and Bright Data, enhances its capabilities. These technologies collectively contribute to the creation of a user-friendly and efficient platform that streamlines the process of tracking and analysing product prices on Amazon. COMPKILLER stands as a testament to the intersection of cutting-edge technology and practical utility, offering users a comprehensive solution to navigate the complexities of e-commerce pricing dynamics.

In conclusion, COMPKILLER addresses the growing demand for real-time e-commerce price tracking by presenting a powerful tool that combines advanced web scraping techniques with a robust technological stack. This project not only serves the immediate need for accurate and timely pricing information on Amazon but also paves the way for future enhancements and expansions, positioning itself as a valuable asset for users navigating the dynamic landscape of online retail.

TABLE OF CONTENTS

1.INTRODUCTION

      1.1 OBJECTIVE
      1.2. SCOPE OF THE PROJECT
      1.3. Motivation For Project

2. BACKGROUND

3. METHODOLOGY

4. TOOLS and TECHNOLOGY USED

5. IMPLEMENTATION

6. CONCLUSION

7. FUTURE ENHANCEMENTS

8. REFRENCES

1.1. OBJECTIVE

The primary objective of COMPKILLER is to revolutionise e-commerce price tracking by providing users with real-time and accurate product pricing information from Amazon. Focused on delivering a seamless user experience, the project aims to empower users through a robust web scraping mechanism, creating and maintaining detailed price histories, and employing advanced technologies for efficiency and reliability. The objective is to ensure users receive timely updates on pricing trends through email notifications, enhancing user engagement. The project's architecture, built on Next.js, Node.js, and Express.js, aims for a responsive and intuitive interface. The incorporation of Bright Data addresses challenges related to site blocking, ensuring consistent and unblocked access to Amazon pages. Looking forward, the project seeks to enhance user authentication, notification systems, UI/UX design, and advanced data analytics, expanding its capabilities to cover additional e-commerce platforms for a more comprehensive product coverage. The objective is to continually improve and innovate, leveraging industry-standard technologies and collaborative efforts within the developer community, as evidenced by the project's transparent source code on GitHub.

1.2. SCOPE OF THE PROJECT

The scope of the COMPKILLER project is extensive, aiming to revolutionise the landscape of e-commerce price tracking by focusing on several key aspects:

Real-time Web Scraping Capabilities:
• COMPKILLER's primary scope is to provide users with up-to-the-minute pricing information from Amazon. The project employs a robust real-time web scraping mechanism that dynamically extracts pricing data from Amazon product pages. This ensures that users receive the latest and most accurate information, contributing to a competitive edge in the fast-paced e-commerce market.

Comprehensive Price History Creation:
• Another significant aspect of the project's scope is the creation and maintenance of a detailed price history for each tracked product. This history includes current, average, lowest, and highest prices, offering users a comprehensive view of pricing trends over time. This feature provides valuable insights for users to make informed decisions based on historical pricing data.

Periodic Data Scraping with Cron Techniques:
• The project employs cron techniques to ensure timely and accurate updates. The periodic data scraping, with a 5-minute frequency, guarantees that users have access to the most recent pricing information. This scope enhances the project's reliability and ensures that users can rely on COMPKILLER for real-time insights into product pricing changes.

Efficient Data Storage with MongoDB:

- COMPKILLER focuses on efficient data storage using MongoDB. Storing scraped data intelligently on a MongoDB server not only facilitates quick data retrieval but also provides a scalable and reliable database solution. This ensures that the project can handle a growing volume of data and maintain its responsiveness over time.

Technological Foundation with Next.js, Node.js, and Express.js:

- The project's core architecture is built on Next.js, Node.js, and Express.js, forming a robust technological foundation. This scope ensures a smooth and responsive user interface, contributing to an overall positive user experience. The choice of these technologies reflects a commitment to staying at the forefront of web development practices.

User Engagement through Nodemailer Integration:

- COMPKILLER enhances user engagement by integrating Nodemailer, enabling the project to send timely email notifications to users based on predefined triggers. This aspect broadens the scope of user interaction and ensures that users are promptly informed about significant price changes, adding value to their experience.

Addressing Site Blocking Challenges with Bright Data:

- To overcome challenges related to site blocking and access restrictions on Amazon, COMPKILLER integrates Bright Data. This comprehensive solution ensures consistent and unblocked access to Amazon pages, with a remarkable 99.9% success rate. This scope enhances the project's reliability and ensures the continuous flow of accurate data.

1.3. Motivation For Project

The motivation behind the development of COMPKILLER stems from the evolving dynamics of the e-commerce landscape and the need for a sophisticated solution to empower users with real-time and comprehensive product pricing information. Several key factors drive the motivation for this project:

Market Competitiveness:
•   In the highly competitive e-commerce market, pricing strategies play a pivotal role in gaining a competitive edge. The motivation for COMPKILLER is rooted in providing users with a tool that enables them to stay ahead of the competition by offering real-time insights into product pricing on Amazon. Timely access to accurate pricing data empowers businesses and consumers alike to make informed decisions and adapt their strategies in response to market fluctuations.

Data-Driven Decision-Making:
•   The project is motivated by the belief in the power of data-driven decision-making. COMPKILLER seeks to provide users with not only current pricing information but also a comprehensive price history for each tracked product. This historical data equips users with valuable insights, allowing them to analyse trends, identify patterns, and make informed decisions based on a thorough understanding of pricing dynamics over time.

User Empowerment:
•   The motivation behind COMPKILLER is centred on user empowerment. By offering a user-friendly interface and timely email notifications, the project aims to empower users, whether they are businesses optimising their

pricing strategies or consumers seeking the best deals. The goal is to enhance the overall user experience by providing a reliable and efficient solution for tracking and understanding product pricing on one of the largest e-commerce platforms.

Technological Innovation:

- The motivation for COMPKILLER is deeply rooted in technological innovation. Leveraging cutting-edge technologies such as Next.js, Node.js, Express.js, MongoDB, Nodemailer, Cheerio, and Bright Data reflects a commitment to staying at the forefront of web development practices. The project is driven by a desire to harness the power of these technologies to create a seamless and responsive solution that meets the evolving needs of users in the digital age.

Addressing Challenges:

- The challenges posed by site blocking and access restrictions on e-commerce platforms, particularly Amazon, motivate the integration of Bright Data. COMPKILLER seeks to overcome these challenges, ensuring consistent and unblocked access to Amazon pages. The project's motivation is rooted in providing a reliable solution that tackles real-world obstacles, enhancing the dependability of the data retrieval process.

2. BACKGROUND

The background of COMPKILLER is grounded in the evolving landscape of e-commerce and the increasing significance of data-driven strategies in the digital marketplace. The project is developed against the backdrop of several key contextual elements:

E-commerce Dynamics:

•   The rapid growth and dynamic nature of the e-commerce sector form a critical background element. As businesses and consumers increasingly rely on online platforms for purchasing and selling products, the need for accurate and real-time pricing information becomes paramount. COMPKILLER emerges as a response to the evolving demands of e-commerce, aiming to provide users with a tool that can navigate and extract crucial pricing data from one of the largest online marketplaces, Amazon.

Data-Driven Decision-Making Trend:

•   The background of COMPKILLER is influenced by the prevailing trend toward data-driven decision-making. Businesses and consumers recognise the strategic advantage of basing decisions on comprehensive and timely data. COMPKILLER aligns with this trend by offering not only real-time pricing information but also a detailed price history, enabling users to make informed decisions based on historical pricing trends and patterns.

Technological Advancements:

•   The backdrop of technological advancements, especially in web development and data processing, plays a pivotal role in shaping COMPKILLER. The project leverages cutting-edge technologies such as Next.js, Node.js, Express.js, MongoDB, Nodemailer, Cheerio, and Bright Data. This technological foundation reflects a commitment to innovation and the utilisation of the latest tools to create a robust and efficient web scraping solution.

Challenges in Web Scraping:

•   The background of COMPKILLER is informed by the challenges inherent in web scraping, especially when dealing with major e-commerce platforms like Amazon. Issues related to site blocking, access restrictions, and the need for consistent and unblocked data retrieval present real challenges. The integration of Bright Data into COMPKILLER is a strategic response to address and overcome these challenges, ensuring a reliable and unimpeded data scraping process.

User Expectations:

•   The evolving expectations of users in the digital era contribute to the background of COMPKILLER. Users, whether businesses or consumers, expect seamless and responsive tools that provide value in real-time. COMPKILLER is designed with a user-centric approach, aiming to meet and exceed these expectations by offering a user-friendly interface, timely email notifications, and a comprehensive set of features for tracking and understanding product pricing on Amazon.

3. METHODOLOGY

The methodology employed in the development and implementation of COMPKILLER involves a strategic and systematic approach to ensure the effectiveness and reliability of the web scraping project. The key steps and methodologies include:

Requirement Analysis:
- The methodology begins with a comprehensive analysis of the requirements, understanding the needs and expectations of users. This involves identifying the key features, data points, and functionalities required for effective e-commerce price tracking on Amazon.

Technology Stack Selection:
- A meticulous process of selecting the appropriate technology stack is crucial. The choice of Next.js, Node.js, Express.js, MongoDB, Nodemailer, Cheerio, and Bright Data is driven by the need for a robust, scalable, and technologically advanced solution. The selected technologies form the foundation of the project's architecture.

System Design and Architecture:
- The methodology includes the design and architectural planning of COMPKILLER. This phase outlines the overall structure of the system, including the database schema, the flow of data, and the integration of various components. The design ensures that the project aligns with best practices in web development and data storage.

Real-Time Web Scraping Mechanism:
- A key aspect of the methodology involves implementing a robust real-time web scraping mechanism. This includes the development of algorithms and scripts that dynamically extract pricing information from Amazon

product pages. The goal is to ensure that users consistently receive the latest and most accurate data.

Price History Creation:

- The methodology includes the implementation of a comprehensive price history creation feature. This involves designing algorithms to capture and maintain detailed price histories for each tracked product. The history includes current, average, lowest, and highest prices, providing users with a comprehensive view of pricing trends.

Cron Technique Implementation:

- To guarantee timely updates, the methodology incorporates cron techniques for periodic data scraping. The system initiates web scraping at regular intervals, with a 5-minute frequency, ensuring that users have access to the most recent pricing information.

MongoDB Data Storage:

- The methodology involves intelligent data storage using MongoDB. This includes designing an efficient database schema to store scraped data. MongoDB provides a scalable and reliable solution for data storage, facilitating quick data retrieval and supporting future scalability.

Nodemailer Integration:

- Methodological steps include integrating Nodemailer to enhance user engagement. The system sends timely email notifications to users based on predefined triggers, ensuring that users are informed about significant price changes.

Cheerio for Data Parsing:

- COMPKILLER employs Cheerio for the intricate task of parsing Amazon product pages and extracting relevant data. The methodology ensures that the parsing process is streamlined for efficiency and accuracy in data extraction.

Bright Data Integration:

- The methodology addresses challenges related to site blocking and access restrictions by integrating Bright Data. This comprehensive solution provides end-to-end site unblocking actions, ensuring consistent and unblocked access to Amazon pages with a high success rate.
- User Interaction and Working of Project:
- Users interact with COMPKILLER by inputting Amazon product URLs. The system, employing cron techniques, initiates periodic web scraping, ensuring timely updates every 5 minutes. The scraped data, including current prices and comprehensive price history, is stored on a MongoDB server. Email notifications keep users informed, and Bright Data ensures consistent and unblocked access to Amazon pages.

4. TOOLS and TECHNOLOGY USED

The development of COMPKILLER involves the use of a carefully selected set of tools and technologies to ensure efficiency, scalability, and effectiveness in real-time web scraping and data processing. The chosen tools and technologies include:

Next.js:

•   Next.js is employed as the foundational framework for COMPKILLER. As a React-based web framework, Next.js facilitates server-side rendering and ensures a responsive and performant user interface. Its modular structure and support for efficient data fetching contribute to a seamless user experience.

Node.js:

•   Node.js serves as the runtime environment for server-side scripting in COMPKILLER. Its non-blocking, event-driven architecture is well-suited for handling asynchronous tasks, such as web scraping and data processing, ensuring optimal performance.

Express.js:

•   Express.js is utilised as the web application framework within Node.js. It simplifies the creation of server-side logic, routing, and middleware integration. Express.js contributes to the development of a robust backend infrastructure for COMPKILLER.

MongoDB:

•   MongoDB is chosen as the database solution for COMPKILLER. Its NoSQL nature allows for flexible and scalable data storage. MongoDB efficiently stores scraped data, including product pricing information and comprehensive price histories.

Nodemailer:

•   Nodemailer is integrated into COMPKILLER to handle email notifications. This technology enables the system to send timely notifications to users based on predefined triggers, enhancing user engagement and keeping users informed about significant price changes.

Cheerio:

- Cheerio is utilised for data parsing in COMPKILLER. This lightweight and fast HTML parsing library simplify the extraction of relevant data from Amazon product pages. Cheerio streamlines the parsing process, contributing to the accuracy and efficiency of web scraping.

Bright Data:

- Bright Data is integrated into COMPKILLER to address challenges related to site blocking and access restrictions on Amazon. This comprehensive solution ensures consistent and unblocked access to Amazon pages, enhancing the reliability of data retrieval with a remarkable 99.9% success rate.

5. IMPLEMENTATION

Project Setup:

- Create a project directory and initialize the project using a version control system, such as Git.
- Set up the development environment, including Node.js and npm for package management.

Frontend Development with Next.js:

- Design and implement the user interface using Next.js, leveraging its React-based components for modularity.
- Create components for user input, display of scraped data, and other relevant features.

Backend Development with Node.js and Express.js:

- Develop the backend server using Node.js and Express.js to handle routing, server logic, and middleware.
- Implement routes for user input, data scraping triggers, and other backend functionalities.

MongoDB Integration:

- Set up a MongoDB database to store scraped data, including product pricing information and price histories.
- Establish a connection between the Node.js backend and the MongoDB database using a MongoDB driver or an Object Data Modelling (ODM) library like Mongoose.

Real-time Web Scraping Mechanism:

- Develop algorithms and scripts for real-time web scraping of Amazon product pages.

- Utilise Cheerio for parsing HTML and extracting relevant pricing data from the scraped pages.
- Implement mechanisms to dynamically fetch and update pricing information at regular intervals using cron techniques.

Email Notification System with Nodemailer:
- Integrate Nodemailer to the backend for sending email notifications to users.
- Define triggers for email notifications based on predefined conditions, such as significant price changes.

Bright Data Integration:
- Configure and integrate Bright Data into the system to handle site blocking challenges on Amazon.
- Implement mechanisms to ensure consistent and unblocked access to Amazon pages, enhancing the reliability of data retrieval.

Testing:
- Conduct thorough testing of the entire system, including frontend components, backend logic, web scraping mechanisms, and email notification functionalities.
- Implement unit tests, integration tests, and end-to-end tests to ensure the robustness and reliability of the application.

User Authentication (Future Enhancement):
- If user authentication is part of future enhancements, implement a secure authentication system to enable personalised dashboards and user-specific tracking preferences.

Enhanced Notification System (Future Enhancement):
- Expand the notification system to incorporate more advanced options and customisable triggers for users to tailor their experience.

UI/UX Improvements (Future Enhancement):

- Continuously improve the user interface design based on user feedback and emerging design trends.

Advanced Data Analytics (Future Enhancement):

- Integrate advanced analytics features to empower users to analyze trends and patterns in product price histories.
- Extend web scraping capabilities to cover additional e-commerce platforms, broadening the scope of product coverage.

Documentation:

- Document the project thoroughly, including code documentation, API documentation, and user guides.
- Ensure that the project's source code, documentation, and other relevant resources are accessible on a version control platform like GitHub.

Deployment:

- Choose an appropriate hosting platform for deployment, considering factors like scalability and performance.
- Deploy the application to a production environment, ensuring that it is accessible to users.

Monitoring and Maintenance:

- Implement monitoring tools to track the performance and reliability of the application.
- Set up a maintenance plan for addressing issues, applying updates, and incorporating user feedback.

6. CONCLUSION

In conclusion, COMPKILLER represents a significant advancement in the realm of e-commerce price tracking, offering users a cutting-edge solution for real-time insights into product pricing on Amazon. The development and implementation of COMPKILLER were guided by a clear vision and strategic methodologies, resulting in a feature-rich and robust web scraping project.

The utilisation of technologies such as Next.js, Node.js, Express.js, MongoDB, Nodemailer, Cheerio, and Bright Data forms a powerful and integrated stack, ensuring a responsive user interface, efficient data storage, and effective web scraping mechanisms. The project's architecture and design prioritise scalability, reliability, and adaptability to the dynamic e-commerce landscape.

Key features, including real-time web scraping, comprehensive price history creation, cron technique implementation for timely updates, MongoDB data storage, and user engagement through Nodemailer, underscore the project's commitment to delivering a seamless user experience. The integration of Bright Data addresses challenges related to site blocking, ensuring consistent and unblocked access to Amazon pages.

Looking forward, the planned future enhancements, such as user authentication, an enhanced notification system, UI/UX improvements, advanced data analytics, and integration with additional e-commerce platforms, demonstrate a forward-thinking approach. These enhancements aim to further empower users, personalise their experience, and provide even more valuable insights for informed decision-making.

The methodology employed in the development process emphasises thorough requirement analysis, careful technology stack selection, and systematic system design. The implementation phase involves meticulous frontend and backend development, integration of external tools, rigorous testing, and documentation to ensure the project's reliability and maintainability.

In summary, COMPKILLER not only addresses the current needs of users in the e-commerce sector but also positions itself for future growth and innovation. The project's transparent source code on GitHub fosters collaboration within the developer community, emphasising openness and transparency. COMPKILLER stands as a testament to the intersection of technology and user-centric design, offering a powerful tool for businesses and consumers navigating the complexities of online product pricing.

7. FUTURE ENHANCEMENT

User Authentication:

- Implementing user authentication will pave the way for personalised dashboards, enabling users to track their specific set of products and preferences.

Enhanced Notification System:

- Future iterations will see an expansion of the notification system, incorporating more advanced options and customisable triggers for users to tailor their experience.

UI/UX Improvements:

- Continuous improvement in user interface design is on the roadmap, focusing on creating a more intuitive and visually appealing experience for users.

Advanced Data Analytics:

- COMPKILLER's future will witness the integration of advanced analytics features, empowering users to analyse trends and patterns in product price histories.

Integration with Additional E-commerce Platforms:

- To broaden its scope, COMPKILLER plans to extend its web scraping capabilities to cover additional e-commerce platforms, offering users a more comprehensive product coverage.

8. REFRENCES

The development of COMPKILLER has been influenced by a myriad of resources from online learning platforms, collaborative efforts with the open-source community, and the adoption of industry-standard technologies. The project's source code is accessible on GitHub, allowing for transparency and collaboration within the developer community.