



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: <https://doi.org/10.22214/ijraset.2024.63128>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Design of Web Applications in ASP.NET

Mr. Anand Talekattu, Dr. Rajani Katiyar, Dr. Sahana B.

Dept. of Electronics and Communication, R V College of Engineering Bengaluru, India

Abstract: *In the digital era, the development of robust web applications is crucial for providing efficient solutions across various domains. This paper presents the design and development of two distinct web applications using ASP.NET Core: a resort booking application and an expense tracker application. The resort booking application utilizes ASP.NET Core Model-View-Controller (MVC) to manage resort operations, while the expense tracker leverages ASP.NET Web API with Angular for comprehensive financial management of the user. The applications are designed to be scalable, secure, and user-friendly, addressing common performance bottlenecks and ensuring optimal performance.*

Index Terms: *ASP.NET Core, Web Application, Resort Booking System, Expense Tracker, MVC, Web API, Angular*

I. INTRODUCTION

The development of web applications has become essential in the modern digital world, providing crucial solutions for various operational needs. ASP.NET Core is a cross-platform framework used for developing end-to-end secure and scalable web applications. Since ASP.NET uses the C#, it is faster than other frameworks that use interpreted language like JavaScript. This paper focuses on the design and development of two web applications: a resort booking system using the ASP.NET Core MVC and an expense tracker using ASP.NET Web API with Angular front-end framework.

II. MOTIVATION AND OBJECTIVE

A. Motivation

Most of the web applications in today's world face computational and performance limitations, such as slow response time and inefficient data processing, which can hinder user experience and operational efficiency. This paper aims to address these limitations by using powerful features of ASP.NET core.

B. Objectives

First objective of the project is to develop web applications that are scalable, provide user authentication and authorization and user-friendly User Interface (UI). Second objective of the project is to automate booking management tasks to reduce manual intervention and streamline operations for resort staff. Third objective is to develop payment gateway to make the payment in resort application. Fourth objective of the project is to develop an expense tracker application that can track expenses and incomes of user with AngularJS.

III. LITERATURE REVIEW

In the dynamic realm of web development, the architecture and tools we choose play a pivotal role in shaping the efficiency, scalability, and security of our applications. From the classic three-tier architecture to cutting-edge frameworks like Vue.js and Angular, each component contributes to the overall success of a project. As the demand for user-friendly interfaces and robust security measures continues to rise, developers are constantly exploring new techniques and methodologies to stay ahead in this ever-evolving landscape. The three-tier architecture, a fundamental approach in web application development highlighted in [1], divides applications into distinct layers: presentation, business logic, and data access. This separation enhances scalability, maintainability, and reduces code coupling. Developers must tailor interfaces between layers based on application needs. As web applications become more prevalent, so do security risks and management costs. Implementing a Web Application Firewall (WAF) like ModSecurity with Reverse Proxy, as demonstrated in [2], effectively blocks various threats including XSS and SQL injection. In [3], a user-friendly WAF framework is proposed, adept at combating common security flaws like XSS, SQLi, and DoS attacks. It simplifies configuration through intuitive interfaces, offers real-time threat detection, and employs regex to filter out malicious code. To secure cloud-based web services, [4] suggests using ontology and identity-based encryption. By distributing data among authorized users, it prevents unauthorized access and maintains confidentiality. [5] compares Vue.js and Angular frameworks for lightweight web application development, focusing on a to-do list application. These frameworks leverage emerging technologies to enhance user experience.

Addressing the challenge of finding compatible web APIs, [6] introduces WAR, a recommendation system based on large-scale API data. It simplifies API discovery and selection, reducing developers’ time and effort. [7] proposes a method to detect SQL injection and XSS attacks without a firewall, utilizing Suricata NIDS/IPS rules and Nginx Reverse Proxy. In [8], a geological exploration fund management system is designed using ASP.NET MVC, Ninject, and jQuery, meeting diverse user needs effectively. [9] evaluates web rendering frameworks—Angular, React, and Vue—based on build size, interaction time, and DOM manipulation. Vue outperforms others in manipulation time. [10] compares Angular and Svelte frameworks based on application complexity and target device, emphasizing the importance of selecting the right framework.

Overall, the web development landscape continues to evolve, offering new solutions to old problems. Choosing the right framework is crucial for developing efficient and secure web applications tailored to specific requirements and business needs.

IV. METHODOLOGY

In project the MVC architecture is followed to build two applications: a resort booking application which automate the booking system by eliminating the human intervention and another is the expense tracker application which is used to track our income and expenses to have set an limit to users daily expenditure.

A. MVC Pattern

Models, Views, and Controllers are the three primary component groups that an application is divided into using the MVC pattern. By following this approach, user requests are sent to a Controller which in response send it to the Model to carry out user actions and obtain query responses. The Controller then ask the View to send the corresponding user interface that the user sees. The model is required by both the controller and the view. However, the model is independent of both the view and the controller. One of the main advantages of the split is that the model can be developed and tested without regard to the visual display because of this separation.

It is easier to code, debug, and test something (a model, view, or controller) that has a single job, hence this division of labor aids in the application’s complexity scaling. Updates, tests, and debugging of code with dependencies across two or more of these three domains are more challenging. For instance, compared to business logic, user interface logic is typically more dynamic. When presentation code and business logic are integrated into one object, each time the user interface is altered, the business logic object needs to be updated. This frequently results in problems and necessitates retesting business logic following even the smallest user interface modification. By keeping each component focused on a single responsibility, the code becomes more modular and easier to manage. Developers can work on the UI without worrying about business logic, and vice versa. Issues can be isolated to a specific layer, making it easier to locate and fix bugs. Each component can be tested independently. For instance, user can write unit tests for the business logic in the Model without involving the UI. When the presentation code and business logic are separated, changes in one do not necessitate changes in the other. This reduces the likelihood of introducing bugs when updating the application. The request flow is shown in figure 1.

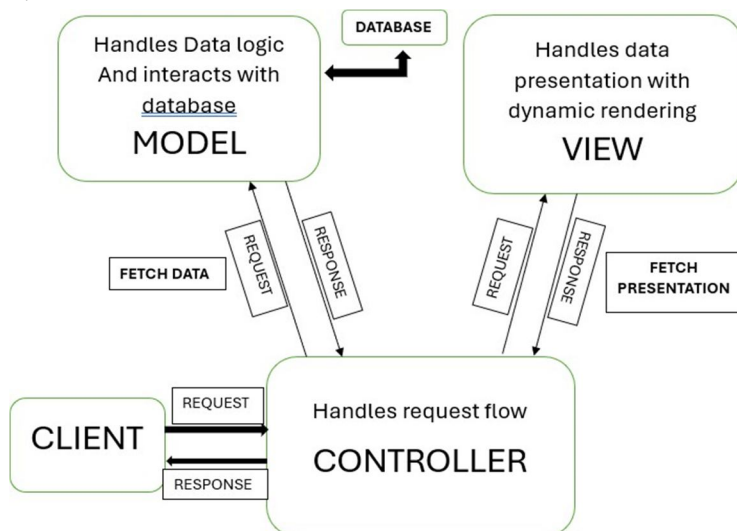


Fig. 1. MVC pattern and data request flow

B. Database Design

The databases for the resort booking and expense tracker application are built in MS SQL server. Figures 4 show the database diagram for the both databases.

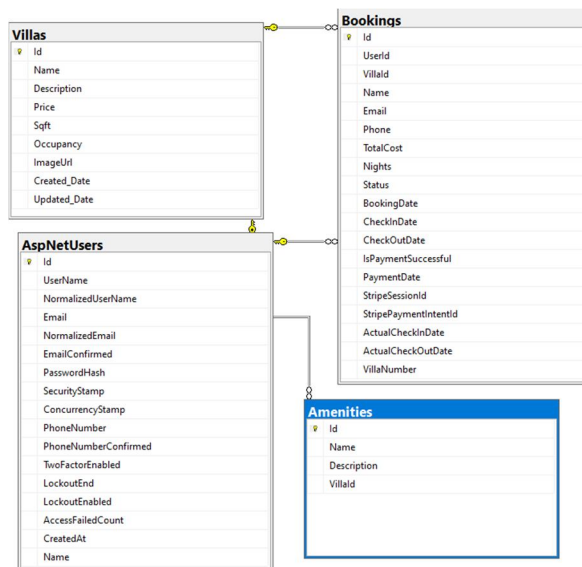


Fig. 2. Resort Booking Application Database

In both applications, the code first approach is used to create a database, entities (tables), and foreign key relationships. Entity Framework introduced the Code First approach with Entity Framework 4.1. The principal application of Code-First is in Domain Driven Design. Instead of designing the database first and then generating the classes that correspond with the database design, developers using the Code-First technique concentrate on the domain of the application and begin creating classes for domain entities.

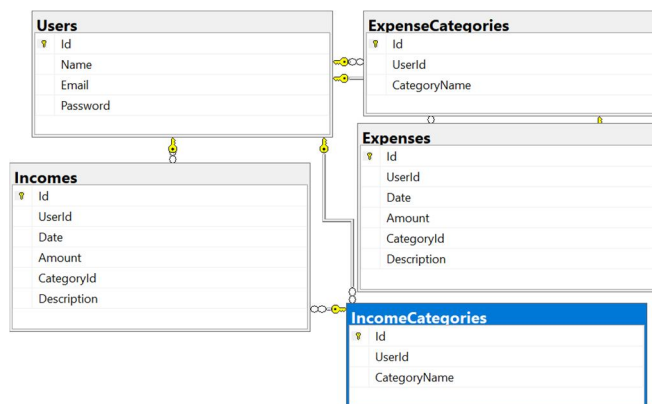


Fig. 3. Expense Tracking Application Database

C. User authentication and authorization

Administrators utilize two essential information security procedures, authorization, and authentication, to safeguard systems and data. Authorization establishes the user or service’s access privileges, whereas authentication confirms the identity of the latter. Authentication is carried out by the register and login process. Authorization is implemented by restricting access to different controllers and actions in source files. In the resort application, there are two roles, one is the admin role and another one is the customer role. Admin can see the dashboard where the data is displayed in graphical format. The admin can add, delete, and update the villa, villa number, and amenities. Admin can see all the bookings and confirm the booking. The customer can only see the available villa and book it. Then he can see the bookings he/she made. In expense tracker there is route guards that are used for restricting the user access to certain pages like navigating to dashboard before login. And user authentication is done using the login and register method where the data can be validated by using FormsModule in angular.

When a user register his name, email, password need to be entered which are validated before storing in database by angular forms module validators. If the entered data are valid then the user is registered and navigated to login page where user has to enter email id and password that he given when registering. If credentials are correct then the user is navigated to dashboard page where expenses and incomes specific to him are displayed

D. Partial page update through partial views

In ASP.NET unlike react the whole page is reloaded whenever the any data in the page is changed. In other frontend frameworks the part of the page which got updated is re-rendered which saves time. So to implement partial page update in ASP.NET MVC framework, partial views are used. In Resort booking application the home page contain the inputs for check in date and number nights the user or customer want to book the villa. Then there is button provided to check the available villa at that date and duration. At this point the whole page is reloaded to get the available villa and display them. But it is not necessary to reload the other components other than the component that display available villa. So prevent reloading of whole page the component that display the available villas is wrapped in an partial view. Then the partial view is called in the home page. Now whenever the user change the check in date and duration of stay and click the search button, only the component that display the available villas is re-rendered

E. API Development

The expense tracker application is developed using the ASP.NET web API. The API endpoints are tested using the postman API testing tool. The API provide flexibility to manipulate data in the client side or the front-end side. The API call is made from the angular service for fetching data. The API calls are handled by the HttpClient module. This service is available as an injectable class, with methods to perform HTTP requests. Each request method has multiple signatures, and the return type varies based on the signature that is called (mainly the values of observe and response type). Figure 4.3 shows the API endpoints. The API endpoints are developed for accessing the expenses and its categories, incomes and its categories, and for login. Each domain has get, post ,put and delete end points.

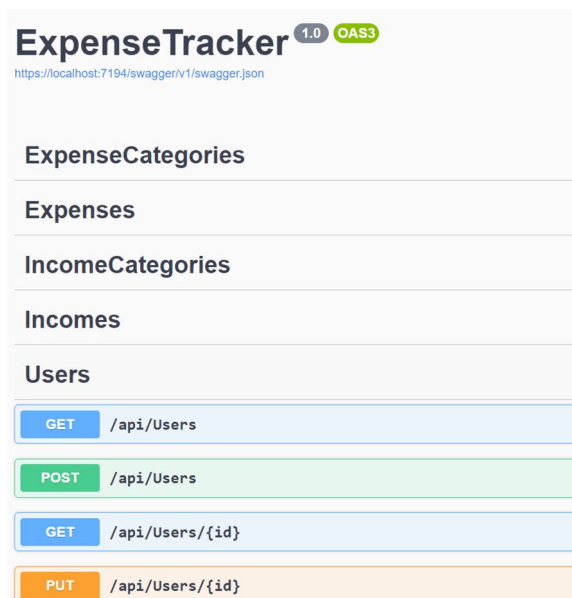


Fig. 4. API end points for Expense tracker

V. RESULTS

The resort booking application will show the increased security through user authentication and authorization. And the clean architecture followed to build the project has the scalability and deployment friendly nature which makes this first priority to large applications. The expense tracker application has the API development which are used when UI is more important than the scalability and speed. It can be used in light weight applications. It also has feature to add banking statement through which income and expense are separated and added to corresponding table. While Angular is

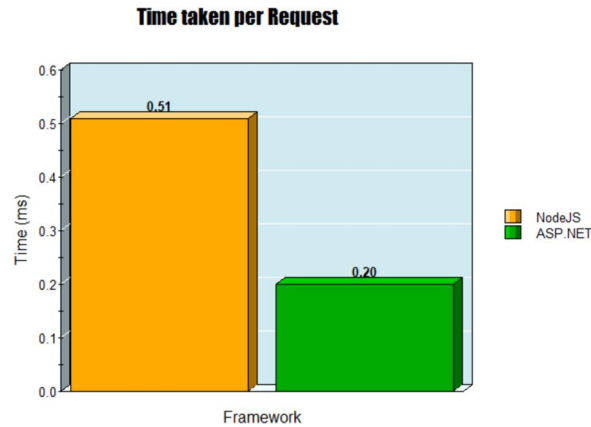


Fig. 5. Time taken per request by ASP.NET and NodeJS

lightweight and ideal for developing single-page applications, it is slower than C# due to JavaScript’s interpreted nature compared to C#’s compiled nature. However, the application’s design mitigates this by offloading heavy data processing to the ASP.NET backend, ensuring a responsive and smooth user experience. The ASP.NET is 11 time faster compared to the NodeJS. Also the .NET version 8 is 18% faster compared to the version 7. The expense tracker UI is designed to provide a seamless and intuitive user experience. It combines robust backend processing with a user-friendly frontend, enabling users to efficiently track and manage their finances. The application’s features, including user authentication, data visualization, and API integration, ensure that it meets the diverse needs of its users while maintaining high standards of performance and security. Comparison of time taken per request by ASP.NET and NodeJS taken per request is analysed by sending 20000 request to MS SQL server. The NodeJS took total 10206.34 ms and the ASP.NET took 3959 ms. It shows the high performance of the ASP.NET core applications and its efficiency in building web API and handling the client request. The comparative analysis of the ASP.NET and NodeJS is shown in figure 5.

REFERENCES

- [1] H. Xing, “Research on .net intelligent platform based on three layer structure,” in 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), IEEE, 2023, pp. 684–688
- [2] R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchi, “Improving security of web-based application using modsecurity and reverse proxy in web application firewall,” in 2020 International Workshop on Big Data and Information Security (IWBIS), IEEE, 2020, pp. 85–90.
- [3] F. Omar, D. Ahmed, O. ElNakib, et al., “Towards a user-friendly web application firewall,” in 2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS), IEEE, 2023, pp. 483–488
- [4] S. Chavan and S. Tamane, “Enhancement in cloud security for web application attacks,” in 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), IEEE, 2020, pp. 91–95.
- [5] O. C. Novac, D. E. Madar, C. M. Novac, G. Bujdos, M. Oproescu, and T. Gal, “Comparative study of some applications made in the angular and vue.js frameworks,” in 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), IEEE, 2021, pp. 1–4
- [6] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, “Data-driven web apis recommendation for building web applications,” IEEE transactions on big data, vol. 8, no. 3, pp. 685–698, 2020.
- [7] P. Tanakas, A. Ilias, and N. Polemi, “A novel system for detecting and preventing sql injection and cross-site-script,” in 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET), IEEE, 2021, pp. 1–6
- [9] P. Luan and G. Li, “Design and implementation of capital management information system for central financial fund project based on mvcpattern,” EAI, Jun. 2023. doi: 10.4108/eai.2-12-2022.2328752.
- [10] R. N. Diniz-Junior, C. C. L. Figueiredo, G. D. S. Russo, et al., “Evaluating the performance of web rendering technologies based on javascript: Angular, react, and vue,” in 2022 XLVIII Latin American Computer Conference (CLEI), IEEE, 2022, pp. 1–9
- [11] T.-D. Tripon, G. A. Gabor, and E. V. Moisi, “Angular and svelte frameworks: A comparative analysis,” in 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), IEEE, 2021, pp. 1–4
- [12] T. F. Iskandar, M. Lubis, T. F. Kusumasari, and A. R. Lubis, “Comparison between client-side and server-side rendering in the web development,” in IOP Conference Series: Materials Science and Engineering, IOP Publishing, vol. 801, 2020, p. 012 136
- [14] K. Peguero and X. Cheng, “Csrf protection in javascript frameworks and the security of javascript applications,” High-Confidence Computing, vol. 1, no. 2, p. 100 035, 2021.
- [15] F. F. Fadlalla and H. T. Elshoush, “Input validation vulnerabilities in web applications: Systematic review, classification, and analysis of the current state-of-the-art,” IEEE Access, vol. 11, pp. 40 128–40 161, 2023.

- [16] V. Veeraiyah, N. B. Rajaboina, G. N. Rao, S. Ahamad, A. Gupta, and C.
- [17] S. Suri, "Securing online web application for iot management," in 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE, 2022, pp. 1499–1504.
- [18] R. Kumar, A. I. Khan, Y. B. Abushark, M. M. Alam, A. Agrawal, and R. A. Khan, "A knowledge-based integrated system of hesitant fuzzy set, ahp and topsis for evaluating security-durability of web applications," IEEE Access, vol. 8, pp. 48 87048 885, 2020.
- [19] R. Kumar, A. I. Khan, Y. B. Abushark, M. M. Alam, A. Agrawal, and
- [20] R. A. Khan, "An integrated approach of fuzzy logic, ahp and topsis for estimating usable-security of web applications," IEEE Access, vol. 8, pp. 50 944–50 957, 2020.
- [21] M. Alenezi, A. Agrawal, R. Kumar, and R. A. Khan, "Evaluating performance of web application security through a fuzzy based hybrid multi-criteria decision making approach: Design tactics perspective," IEEE Access, vol. 8, pp. 25 54325 556, 2020.
- [22] P. Kalariya, M. Jethva, and Y. Alginahi, "MI assisted web application firewall," in 2024 12th International Symposium on Digital Forensics and Security (ISDFS), IEEE, 2024, pp. 1–6.
- [23] I. Ahmad, E. Suwarni, R. I. Borman, F. Rossi, Y. Jusman, et al., "Implementation of restful api web services architecture takeaway application development," in 2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS), IEEE, 2021, pp.132–133
- [24] A. Ansari, P. Lotfi-Kamran, and H. Sarbazi-Azad, "Divide and conquer frontend bottleneck," in 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), IEEE, 2020, pp. 65–78.
- [25] P. Pant, A. S. Rajawat, S. Goyal, et al., "Authentication and authorization in modern web apps for data security using nodejs and role of dark web," Procedia Computer Science, vol. 215, pp. 781–790, 2022.
- [26] R. F. Olanrewaju, B. U. I. Khan, M. A. Morshidi, F. Anwar, and M. L.
- [27] M. Kiah, "A frictionless and secure user authentication in web-based premium applications," IEEE Access, vol. 9, pp. 129 24129 255, 2021.
- [28] J. V. Chandra, N. Challa, and S. K. Pasupuletti, "Authentication and authorization mechanism for cloud security," International Journal of Engineering and Advanced Technology, vol. 8, no. 6, pp. 2072–2078, 2019. UG Major Project Report Department of ECE, 2023-24 45.
- [29] S. A. Sabab, S. S. Islam, M. J. Rana, and M. Hossain, "Eexpense: A smart approach to track everyday expense," in 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), IEEE, 2018, pp. 136–141.
- [30] E. Johri, P. Desai, P. Soni, H. Jain, and N. Sanganageria, "Expense management system," in 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), IEEE, 2023, pp. 1–6.
- [31] N. Upadhaya, A. Pant, S. Karthikeyan, and A. L. S, "Web automation using selenium in java," in International Research Journal of Engineering and Technology, IRJET, vol. 07, May 2020, pp. 6011–6014. UG Major Project Report Department of ECE, 2023-24 53
- [32] S. B. A. P. R, C. M. M. Kumar, S. B, G. B, and A. Deshpande, "Implementation of hybrid load balancing algorithm in web servers," in International Journal of Advanced Science and Technology, IJAST, vol. 29, Feb. 2020, pp. 2234–2241.
- [33] P. Biswagar SiddharthaBhaumik, "Detection and classification of malicious websites," in International Research Journal of Modernization in Engineering Technology and Science, IRJMET, vol. 03, May 2021, pp. 1–6.
- [34] A. Dinkar and P. Biswagar, "Analysis of data oriented web application systems," in Journal of University of Shanghai for Science and Technology, vol. 23, May 2021, pp. 636–649. doi: 10.51201/JUSST/21/05199.
- [35] N. G. ANUSHA LS, "Inventory management website for maintenance of the corporate computer equipment's," in International Research Journal of Modernization in Engineering Technology and Science, IRJMET, vol.03, May 2021, pp. 1–4.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)