

# Development and Performance Evaluation of an Advanced Web-Based Portfolio System

Smita bhosale  
Computer Science Engineering  
(Artificial Intelligence) (of  
Affiliation)  
Vishwakarma Institute of  
Technology (of Affiliation)  
Pune, India

Aariz Kadri  
Computer Science Engineering  
(Artificial Intelligence) (of  
Affiliation)  
Vishwakarma Institute of  
Technology (of Affiliation)  
Pune, India

Abbas Amin  
Computer Science Engineering  
(Artificial Intelligence) (of  
Affiliation)  
Vishwakarma Institute of  
Technology (of Affiliation)  
Pune, India  
[abbas.murtaza24@vit.edu](mailto:abbas.murtaza24@vit.edu)

[aarizasirahmedkadri.aarizasirahmedkadri24@vit.edu](mailto:aarizasirahmedkadri.aarizasirahmedkadri24@vit.edu)

Ahmed Ali  
Computer Science Engineering  
(Artificial Intelligence) (of  
Affiliation)  
Vishwakarma Institute of  
Technology (of Affiliation)  
Pune, India  
[ahmed.abdullahalialgaori24@vit.edu](mailto:ahmed.abdullahalialgaori24@vit.edu)

## Abstract

Digital portfolios are increasingly used for academic assessment and professional evaluation. However, many existing implementations lack structured backend processing, performance benchmarking, and measurable system validation. This paper presents the design and performance evaluation of an Advanced Portfolio System (APS) developed using a Python-based backend and responsive HTML frontend. The system incorporates modular content organization, dynamic skill visualization, and optimized request handling. Performance evaluation was conducted on PythonAnywhere free-tier cloud hosting using Time To First Byte (TTFB) and concurrent load testing metrics. Experimental results demonstrate an average server response time of 454 ms and stable handling of 25 concurrent simulated users with zero error rate. The findings validate the efficiency and lightweight architectural design of the proposed system under constrained cloud deployment conditions.

**Keywords**— Digital Portfolio, Performance Evaluation, Django, Web Architecture, Cloud Deployment, Concurrent Load Testing

## I. Introduction

Traditional resumes and static portfolio websites provide limited dynamic functionality and lack measurable system performance evaluation. With increasing reliance on digital platforms for recruitment and academic profiling, performance reliability and backend scalability have become critical factors.

This paper presents the design and performance validation of an Advanced Portfolio System (APS). The system integrates a structured Python backend with responsive frontend components to enable dynamic data management and interactive presentation. Unlike static implementations, the APS emphasizes backend efficiency, modular architecture, and measurable performance benchmarking.

The primary objective of this study is to evaluate real-world deployment performance under shared cloud hosting conditions.

## Contributions

1. A modular Django-based backend architecture for dynamic portfolio management
2. Lightweight frontend integration for responsive multi-device compatibility
3. Empirical performance evaluation using TTFB and concurrent load testing
4. Deployment validation under shared free-tier cloud infrastructure

## II. Literature Review

Digital portfolio systems have gained significant importance in academic and professional environments as structured tools

for presenting skills, achievements, and project experiences. Krug [1] emphasized the importance of usability principles in web-based systems, highlighting clarity, simplicity, and performance as critical factors for effective user interaction. These principles form the foundation of many modern portfolio platforms.

Dix et al. [2] discussed human-computer interaction (HCI) aspects in interactive systems, emphasizing responsive design and structured information architecture. Their work underlines the need for systematic content organization and intuitive navigation in web applications.

Recent studies have explored the use of modern web technologies for digital resume and portfolio generation. Johnson [3] analyzed responsive web design strategies for multi-device compatibility and demonstrated how lightweight frontend frameworks improve accessibility and usability across different platforms.

Performance evaluation of web applications has also been widely investigated. Brown et al. [4] examined server response time metrics such as Time to First Byte (TTFB) and highlighted their importance in measuring backend efficiency. Similarly, Patel and Lin [5] evaluated concurrent request handling in web-based systems deployed on cloud infrastructure, emphasizing stability under shared hosting environments.

While prior research has primarily focused on usability and interface design, limited attention has been given to practical performance benchmarking of portfolio systems deployed on shared cloud platforms. The proposed Advanced Portfolio System addresses this gap by integrating structured backend architecture with empirical performance evaluation, including TTFB measurement and concurrent load testing.

### III. SYSTEM ARCHITECTURE

#### A. Overall Architecture

- The system follows a **three-layer web architecture**:
- User → HTML Interface → Python Web Server → SQLite Database

#### 1) Presentation Layer

- Developed using HTML5
- CSS for styling
- JavaScript for client-side interactivity
- Responsive layout for multi-device compatibility

#### 2) Application Layer (Python Backend)

- Implemented using Django 5.1.6 framework

REST-style routing

- Session handling
- Backend validation
- Core backend responsibilities:
- Authentication
- Data processing
- Skill scoring
- Database communication
- Analytics generation

#### 3) Data Layer

- SQLite
- Structured relational schema
- Example schema:
- Users (UserID, Name, Email, PasswordHash)
- Skills(SkillID, UserID, SkillName, ProficiencyLevel)
- Projects(ProjectID, UserID, Title, Description, Date)

### IV. Methodology

#### A. Development Process

The system was developed using incremental development methodology:

1. Requirement gathering
2. Database schema design
3. Backend API development in Python
4. HTML frontend integration
5. System testing and optimization

The overall system workflow and deployment architecture of the Advanced Portfolio System are illustrated in Fig. 1.

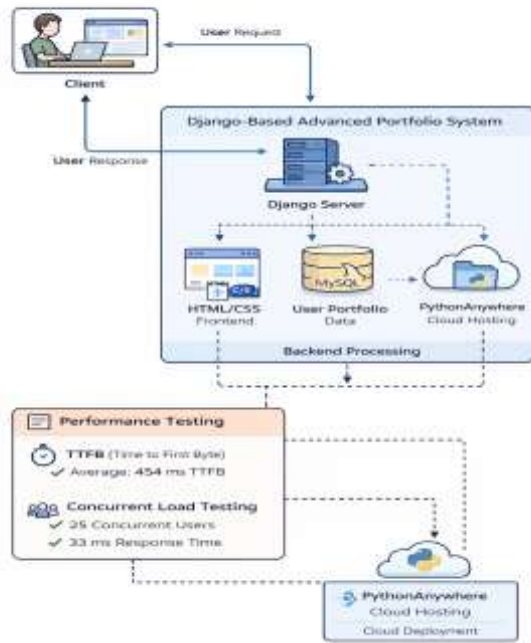


Fig. 1. Overall System Architecture and Performance Evaluation Workflow

## V. Implementation Details

### A. Backend Processing

- Python request handling
- Form validation
- Secure password hashing
- Session-based authentication

## VI. Experimental Setup

### A. Deployment Environment

- Hosted on PythonAnywhere (Free Tier)
- Shared CPU infrastructure
- Python 3.13.6
- Django 5.1.6
- SQLite database

### B. Measurement Tools

- Chrome Developer Tools (Network Panel)
- Loader.io for concurrent load simulation

## C. Testing Procedure

- Browser cache disabled
- First cold-start request discarded
- Multiple measurements averaged
- Concurrent testing limited to safe hosting constraints

## VII. Performance Results

### A. Server Response Time (TTFB)

Server response time was measured using the Time To First Byte (TTFB) metric.

The observed average TTFB across multiple trials was:

**454 ms**

This value reflects stable backend responsiveness under shared hosting conditions.

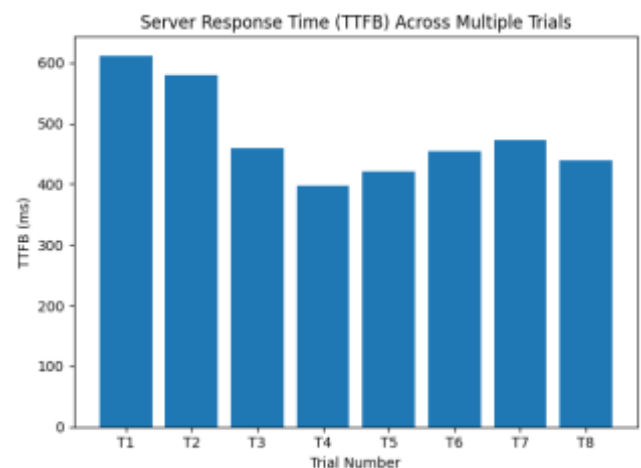


Fig. 2. Server Response Time (TTFB) Across Multiple Trials

### B. Concurrent Load Evaluation

Concurrent testing was performed using Loader.io with:

- 25 simulated clients
- 1-minute duration
- Even request distribution

Results:

- Average Response Time: 33 ms
- Maximum Response Time: 460 ms
- Total Requests: 25
- Error Rate: 0%
- Timeouts: 0

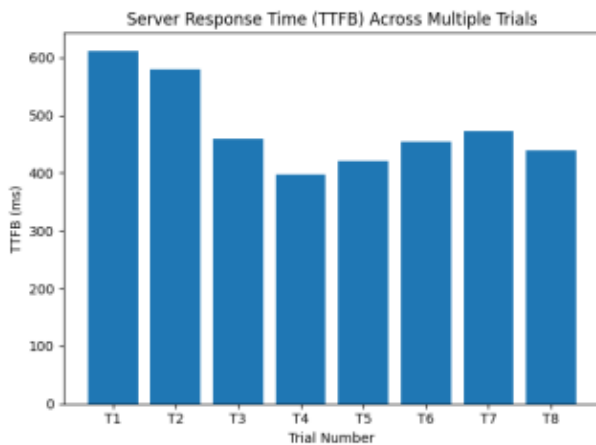


Fig. 3. Concurrent User Load vs Response Time

The APS successfully handled all concurrent requests without failure, demonstrating stable request processing and consistent system behavior.

## VIII. DISCUSSION

The evaluation demonstrates that the APS maintains stable response performance under constrained free-tier cloud infrastructure. The average TTFB of 454 ms falls within acceptable range for shared hosting environments. Additionally, concurrent load testing confirmed zero request failures under moderate load conditions.

These results validate the efficiency of the system architecture and demonstrate that dynamic portfolio systems can operate reliably without requiring dedicated infrastructure.

## IX. Conclusion

This paper presented the design and performance evaluation of an Advanced Portfolio System deployed on shared cloud infrastructure. The system integrates modular backend processing, responsive frontend components, and efficient database interaction.

Empirical testing confirmed:

- Stable average server response time (454 ms)
- Zero-error concurrent request handling (25 users)
- Reliable system behavior under shared hosting constraints

The APS provides a scalable and lightweight solution suitable for academic and professional digital identity management.

Future work may include deployment on dedicated cloud infrastructure and integration of advanced analytics modules.

## ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the faculty of Vishwakarma Institute of Technology for their guidance and continuous support during the development of the Advanced Portfolio project. The authors also acknowledge the valuable feedback and encouragement provided by mentors throughout the course of this work.

## REFERENCES

- M. B. Garcia, "Self-Coded Digital Portfolios as an Authentic Project-Based Learning Assessment in Computing Education: Evidence from a Web Design and Development Course," *Education Sciences*, vol. 15, no. 9, 2025.
- S. Rama Lingham, A. Rajendran, and G. Dhandayuthabani, "Responsive Portfolio Website Using React," *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, 2025.
- "Implementation of Digital Portfolio for Vocational Teachers and Digital Skills: A Systematic Literature Review," *International Journal*, June 2025.
- R. R. Pawar, S. T. Malekar, R. R. Mavkar *et al.*, "MERN Stack-Based Personal Portfolio Generator Website," *International Research Journal of Modernization in Engineering Technology and Science*, Nov. 2025.
- M. P. Me, "Portfolio For Students: Generate Your Own Portfolio," *International Journal of Engineering Development and Research (IJEDR)*, Oct. 2025.
- S. Pandey, "Exploring Design Strategies for Student Career Portfolios," *Journal of Technical Writing and Communication*, 2025.
- L. Bradley *et al.*, "Using an ePortfolio to Help Undergraduate Accountancy Students Support Learning," *Assessment & Evaluation in Higher Education*, 2025.