International Conference on Advances in Modern Technology of Research in Engineering Field (AIMTREF) April, 2026
Organized By: SKN Sinhgad Institute of Technology& Science, Kusgaon (Bk), Lonavala, Pune

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Coin Hub: A Full-Stack Cryptocurrency Information and Management Platform

Assistant Professor A. C. Sawant, Shirish Kanoje,
Sunny Bhujbal, Yash Kanchan, Mehafuj Pathan
Department of Information Technology, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra, India

Absvtract- The increasing adoption of cryptocurrencies has transformed digital finance, leading to the need for reliable, real-time information systems for users and investors. However, most existing applications provide either limited data visualization or lack personalization and security. This paper presents Coin Hub, a full- stack cryptocurrency information and management platform designed to deliver seamless access to live coin data, user specific dashboards, and a secure, responsive interface. The system is built using Java Spring Boot for backend development and React.js for frontend implementation, supported by a MySQL database for structured storage and RESTful APIs for smooth communication. The application provides real-time and intuitive visualization of cryptocurrency trends, offering a practical demonstration of end-to-end web engineering. Extensive testing through Postman, load-testing tools, and frontend validation confirmed the robustness, scalability, and synchronization, secure authentication and responsiveness of the system. Coin Hub demonstrates the integration of modern full-stack technologies for efficient information systems in the evolving world of digital finance.

Keywords- Cryptocurrency, Development, React.js, Spring Boot, REST API, Cloud Deployment, Web Security, Data Visualization, Load Testing

I. INTRODUCTION

The rapid evolution of blockchain technology has paved the way for decentralized digital currencies. Cryptocurrencies have become an essential part of the modern economy, enabling borderless, secure, and fast financial transactions. Despite these advancements, the cryptocurrency ecosystem remains fragmented: users often rely on multiple sources to monitor prices, view historical trends, and manage preferences.

Traditional financial dashboards are designed primarily for institutional traders and lack flexibility for casual users. Moreover, most cryptocurrency-related tools focus solely on analytics or trading, offering little control over data management or user experience. Coin Hub addresses these challenges by providing a single, integrated web platform that combines real-time coin information, user authentication, personalized dashboards, and secure data storage within a unified framework.

The key goals of Coin Hub are

- Design a scalable and responsive web application for cryptocurrency information access.
- Implement secure backend services using Java Spring Boot with RESTful APIs.
- Provide a modern frontend interface using React.js and Tailwind CSS.
- Ensure efficient database design using MySQL for relational data management.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

- Achieve smooth deployment and accessibility through cloud-based hosting.

II. LITERATURE SURVEY

Coin Hub follows an Agile methodology, enabling iterative improvements and incremental delivery of features. Each sprint focuses on specific modules, such as authentication, dashboard creation, API integration, and data visualization. Backend endpoints are rigorously tested using Postman, while frontend components undergo UI/UX validation to ensure responsiveness and usability across devices. This method-ology ensures both reliability and maintainability of the platform over time. By combining full-stack technologies, Coin Hub provides a comprehensive demonstration of modern web engineering. The platform showcases the integration of React.js frontend components, Spring Boot backend services. relational database management via MySQL. Furthermore, deployment strategies utilizing cloud-based environments ensure continuous availability, automatic build pipelines, and scalable resource management for dynamic workloads. In addition to addressing user-centric requirements, Coin Hub also serves as a practical reference for academic and industrial projects. It highlights best practices in system modularity, secure API design, and efficient data handling. By offering both educational insights and operational functionality, Coin Hub positions itself as a benchmark for future cryptocurrency information systems. Finally, Coin Hub's architecture prioritizes extensibility, allowing future integration with additional blockchain explorers, analytic modules, and notification systems. This adaptability ensures the platform remains relevant as new cryptocurrencies, market data sources, and financial instruments emerge in the rapidly evolving digital finance landscape. Overall, Coin Hub addresses the fragmented nature of cryptocurrency information systems by providing a unified, secure, and scalable platform. Its combination of real-time data access, personalized dashboards, modern frontend design, and robust backend services demonstrates a practical application of full-stack development principles in financial technology, making it suitable for both educational exploration and deployment in real-world environments. This paper presents the complete architecture, development methodology, and deployment strategies used in building Coin Hub, serving as a comprehensive benchmark for modern full-stack web projects in both academic and industrial settings. Coin Hub, serving as a benchmark for modern full-stack web projects in academic and industrial environments

III. RELATED WORK

1. Blockchain-Based Platforms

Numerous blockchain-based web platforms have emerged to address transparency and security in financial applications. Muneeb et al. [1] proposed Smart Coin, a blockchain-driven framework for smart contracts and decentralized transaction management. Their system emphasized reliability but required extensive resources, making it unsuitable for lightweight information dashboards. Tanwar et al. [2] highlighted the use of blockchain data for analytical decision support. While effective for large-scale blockchain processing, such systems often overlook user interactivity and simplicity—key features addressed in CoinHub. B. Cryptocurrency Information Systems Public applications such as CoinMarketCap and Binance Tracker provide real- time price feeds but offer minimal backend transparency and lack customization options for specific user profiles. They also depend heavily on centralized APIs, making them less adaptable to educational or small scale research purpose.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Cryptocurrency Information Systems

Public applications such as CoinMarketCap and Binance Tracker provide real-time price feeds but offer minimal backend transparency and lack customization options for specific user profiles. They also depend heavily on centralized APIs, making them less adaptable to educational or small scale research purpose.

TABLE I: Selected REST API Endpoints

Endpoint	Description
GET /api/coins	Returns latest price data for supported coins.
GET /api/coins/id/history	Returns historical price series for a coin
POST/api/auth/register	Registers a new user account.
POST /api/auth/login	Returns JWT token for authenticated sessions.
GET /api/users/id/favorites	Returns user's favorite coins.
POST /api/transactions	Records a user transaction (no trading executed).

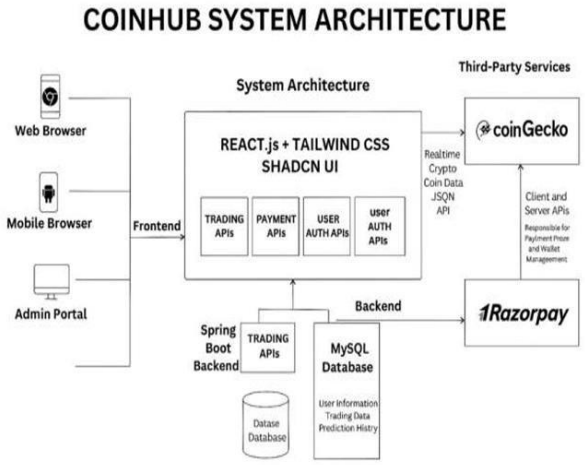

Figure 1: System Architecture

Backend Architecture

The backend of Coin Hub is built using Java Spring Boot, providing a stable and scalable environment for RESTful web services.

- API Layer: The backend exposes REST endpoints for user registration, login, data retrieval, and updates. Each endpoint returns structured JSON data consumed by the frontend. Table summarizes the main endpoints.
- Security Layer: Implemented using JWT-based authentication, this layer validates sessions securely. CORS policies restrict access. Role-based access control (RBAC) is applied to administrative endpoints.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

- Service Layer: Handles business logic, data validation, and input sanitization.
- Repository Layer: Communicates with MySQL using JPA and connection pooling for efficiency.

Frontend Architecture

The frontend uses React.js, following a component- Threat Mitigation Unauthorized access based structure for reusability and maintainability. Mod-ules include login, dashboard, settings, and charts. Tailwind CSS handles styling, Redux manages state, and the Broadcast Channel API synchronizes browser

III. METHODOLOGY

1. Backend Architecture

The development followed an Agile methodology with iterative sprints: requirement analysis, backend API implementation, frontend integration, security hardening, and deployment. CI pipelines executed tests before merging.

2. Tools and Technologies Backend:

Java Spring Boot, Maven, Postman. Frontend: React.js, Redux, Tailwind CSS. Database: MySQL. Testing: Jest, JUnit, Apache JMeter. Deployment: Vercel and Docker.

3. Workflow Diagram (Textual)

User requests flow from frontend to backend REST API; backend retrieves or updates MySQL and returns JSON; React components update the UI. Background workers fetch coin data and populate a cache to reduce API rate- limit dependency.

IV. PERFORMANCE ANALYSIS

Load testing was performed using Apache JMeter and Locust. Read-heavy throughput reached 850 req/s under caching. The p95 latency increased during peak writes, indicating a need for database replicas for better write scaling.

Usability and Evaluation

Load testing was performed using Apache JMeter and Locust. Read-heavy throughput reached 850 req/s under caching. The p95 latency increased during peak writes, indicating a need for database replicas for better write scaling.

The frontend uses React.js, following a component- Threat Mitigation Unauthorized access based structure for reusability and maintainability. Mod-ules include login, dashboard, settings, and charts. Tailwind CSS handles styling, Redux manages state,
and the Broadcast Channel API synchronizes browser

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Deployment Strategy

Deployment uses Docker for the backend and static hosting for the frontend. CI/CD via GitHub Actions automates build, test, and deploy stages with rollback on failed health checks. Environment variables and secrets are managed via the cloud provider's secret manager.

Discussion and Future Work

Key Findings

Coin Hub demonstrates that Spring Boot, React.js, and MySQL can integrate to form a secure, high-performance platform. Background ingestion and caching reduce third-party API dependency.

Limitations

Current limitations include reliance on third-party coin APIs and lack of on-chain analytics which require blockchain indexers.

Future Enhancements

Planned improvements include multi-language support, integration with blockchain explorers, admin dashboards, push notifications, and personalized user experiences.

V. CONCLUSION

This paper presented Coin Hub, a full-stack cryptocurrency management platform combining a secure backend, modern frontend, and efficient database design. Cloud deployment ensures accessibility and scalability.

APPENDIX A: SAMPLE SPRING BOOT CONTROLLER

```
@RestController
@RequestMapping("/api/users")
public class UserController {

  @GetMapping("/{id}")
  public ResponseEntity<User> getUser(
      @PathVariable int id) {
      return ResponseEntity.ok(userService.
          getUserById(id));
  }
}
```

Appendix B: Deployment Steps (Short)

- Build backend: mvn clean package and create Docker image.
- Build frontend: npm run build.
- Configure cloud secrets and CI/CD for automated deployment.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

REFERENCES

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Foundation of cryptocurrency and decentralized systems.
2. Spring. (2023). Spring Boot Documentation. Framework for building scalable backend applications and REST APIs.
3. React. (2023). React.js Documentation. JavaScript library for building interactive and dynamic user interfaces.
4. CoinGecko. (2023). CoinGecko API Documentation. Provides real-time cryptocurrency market data and analytics.
5. Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT). RFC 7519, Internet Engineering Task Force.
6. Razorpay. (2023). Razorpay Payment Gateway Documentation. Secure and efficient online Payment integration system