Infosys Springboard Virtual Internship 6.0 Completion Report

Details :

Name : Kazi Jiya Mukim

Batch Number : 10

Start date : 25-nov-25

End date : 26-feb-26

Internship Duration: 8 WeeK

- **Project Title**

Smart Home Energy Management System

- **Project Objective**

The primary objective of the Smart Home Energy Management System is to design and implement a scalable, secure, and user-friendly platform that monitors, analyzes, and optimizes household energy consumption. The system should allow homeowners to visualize consumption patterns, control compatible devices remotely, schedule automations for energy savings, and receive actionable recommendations to reduce peak loads and overall bills. Emphasis is placed on modular architecture, RESTful APIs, and an intuitive web interface suitable for future mobile extension.

- **Project Description**

This project implements a full-stack web application using Java 17 and Spring Boot for server-side services, backed by a relational database (MySQL or PostgreSQL) and a modern responsive front end with HTML5, CSS3, and Bootstrap. Core features include device registration, real-time and historical consumption dashboards, rule-based automations, user authentication and role-based access, and REST APIs for third-party integrations. The backend integrates Spring MVC and Spring Data JPA (or Hibernate) for data persistence and transaction management. Build automation is handled with Maven; development best practices were followed using Git and GitHub for version control. Postman was used to design and test APIs. The system focuses on energy efficiency metrics, peak detection, scheduled device control, and an extensible plugin model for adding new device types or analytics modules.

- **Timeline Overview (8 Weeks)**

Week	Activities Planned	Activities Completed
Week 1	Project kickoff, requirement gathering, architecture design, environment setup	Kickoff completed, requirements finalized, repo created, basic Spring Boot skeleton established
Week 2	Database schema design, authentication module, basic REST endpoints	DB schema implemented, Spring Security scaffolded, user CRUD APIs functional
Week 3	Device model, device registration APIs, initial frontend wireframes	Device entities & APIs built, frontend wireframes completed, Bootstrap scaffold integrated
Week 4	Real-time data ingestion prototype, dashboard design, API tests	Simulated ingestion implemented, dashboard layout developed, Postman collections created
Week 5	Historical analytics, reporting, rules engine prototype	Analytics queries added, report endpoints available, simple rules engine implemented
Week 6	UI polish, responsiveness, automation scheduling, integration testing	UI refined with Bootstrap, scheduling endpoints done, integration tests executed
Week 7	Performance tuning, security review, documentation drafting	Indexes added, security review completed, documentation draft started
Week 8	Final testing, deployment scripts, presentation and report preparation	Final tests passed, deployment instructions documented, presentation prepared

- ## Key Milestones

Milestone	Description	Date Achieved
Project Kickoff	Initial project briefing, objective finalization, and research on customer feedback and sentiment analysis techniques.	28-nov-25
Prototype / First Draft	Completion of data collection, preprocessing, and development of the initial sentiment analysis and recommendation model.	10-dec-25
Mid-Term Review	Review of model performance, team formation evaluation results, dashboard progress, and discussions on enhancements for the final system.	12-jan-26
Final Submission	Completion of model training, dashboard/interface implementation, documentation, and system deployment for submission.	28-jan-26
Presentation	Final demonstration of the project showing sentiment insights, recommendations, and full dashboard functionality.	31-jan-26

- **Project Execution — Backend Technologies**

The backend was implemented using Java 17 and Spring Boot as the foundational framework. Core responsibilities included designing a modular, maintainable service layer and exposing secure REST endpoints for all client interactions. Spring MVC provided controller abstraction and request handling, while Spring Security handled authentication and role-based access. For data persistence, Spring Data JPA (or Hibernate) was used to map domain entities to the relational database. Transactions and connection pooling were configured to ensure data integrity and performance.

Java 17

Leverage of modern language features to improve code clarity and performance.

Spring Boot

Auto-configuration, dependency management, and production-ready defaults accelerated development.

Spring MVC & REST APIs

Clear routing, DTOs, and API versioning patterns established for future extensibility.

Spring Data JPA / Hibernate

Repository patterns and JPQL queries implemented for analytics and historical queries.

Maven & CI

Build and dependency management with Maven; CI pipelines defined for automated tests and packaging.

- ## Project Execution — Frontend & Integration

The frontend was built with HTML5, CSS3, and Bootstrap to deliver a responsive, accessible UI. Key pages included the authentication screens, device management console, real-time dashboard, and analytics reports. Emphasis was placed on clarity, mobile-first responsiveness, and chart-driven visualizations for energy metrics. RESTful APIs exposed by the backend were consumed by the frontend using fetch patterns; Postman collections documented the endpoints for testing and integration. End-to-end flows were validated through integration testing and manual exploratory testing.

The frontend for the energy project was developed using HTML5, CSS3, and Bootstrap, focusing on a responsive, mobile-first design for authentication, device management, and analytics dashboard pages. RESTful APIs were consumed via fetch , with Postman ensuring robust integration and validation

Dashboard

Real-time summaries, peak load indicators, and quick actions to toggle devices or set schedules.

Device Management

Register devices, assign rooms, and set energy profiles for each device type.

Analytics & Reports

Historical consumption charts, exportable CSVs, and recommendations for optimization.

- ### Learnings & Skills Acquired

Throughout the internship, team members acquired and strengthened both technical and soft skills. Technically, participants gained hands-on experience with Java 17 features, Spring Boot application design, REST API development, JPA mapping, and relational database modeling. Frontend skills included responsive UI design with Bootstrap, accessibility considerations, and client-side integration with backend services.

Courses Completed

Multiple targeted modules on Java, Spring, and web development.

Certifications

Program certifications validating project-based competencies.

Project Duration

Full lifecycle delivery across 8 weeks with iterative sprints.

Soft skills

improved communication, structured documentation practices, version control collaboration (feature branching), test-driven thinking, and presentation skills during milestone demos. The team also practiced time-boxing, backlog prioritization, and risk mitigation strategies when facing integration issues.

- ## Challenges Faced

During the project lifecycle the team encountered several technical and process challenges that were addressed iteratively. Key technical challenges included integrating simulated real-time device telemetry with the stateless HTTP API model, balancing query complexity for analytics against database performance, and implementing secure scheduling for device automations. Platform-related constraints—such as limited access to certain cloud resources for deployment—required local containerized testing and careful deployment documentation.

Telemetry Simulation

Created reliable simulators for device data to test dashboards and analytics without physical hardware.

Database Performance

Introduced indexing, query optimization, and bounded aggregation windows to keep analytics responsive.

Scheduling & Security

Ensured scheduled tasks execute securely by validating authorization for automation actions and using job queues.

Collaboration

Coordinated code merges and handled conflicting changes through frequent syncs and clear branching strategies.

- **Testimonials from Team**

Below are condensed testimonials capturing individual perspectives on the internship experience and team collaboration.

Lead Developer

"This internship accelerated our understanding of production-ready Java systems and taught practical trade-offs between implementation speed and maintainability."

Backend Engineer

"Working with Spring Data JPA and designing analytics queries deepened my database design skills and appreciation for profiling."

Frontend Engineer

"Building the responsive dashboard reinforced the importance of UX-first thinking when presenting technical data to end users."

QA & Documentation

"Testing integration flows and documenting Postman suites improved the reproducibility of our demo and handover materials."

Conclusion

Conclusion: The Smart Home Energy Management System project, completed as part of the Infosys Springboard Virtual Internship 6.0, successfully demonstrated the team’s ability to design and deliver a full-stack, extensible solution within the 8-week timeline. The project met its primary objectives: device management, energy visualization, analytics, scheduling automations, and well-documented REST APIs. The experience showcased how structured virtual internships can bridge academic learning and practical software engineering skills, preparing participants for real-world product development.

Acknowledgements

I express gratitude to Infosys Springboard mentors, technical reviewers, and program coordinators for guidance and support. Special thanks to peer reviewers who performed test cycles and helped refine the deliverables. The team looks forward to continuing development, incorporating user feedback, and contributing open-source components back to the community.