

Fluento Web Application

1. Project Overview

The application is designed to help users—such as newcomers, students, and global workers—learn new languages through AI-assisted tutoring, structured lessons, and gamified learning exercises.

2. Business Objectives

- Provide an engaging language learning experience through interactive lessons.
- Evaluate the feasibility of integrating AI-based tutoring.
- Compare two technology stacks across performance, and scalability.

3. Hypotheses

1- Using an AI-based tutor will improve user engagement by at least 30% compared to traditional static lessons.

2- A Spring Boot + PostgreSQL backend offers better performance and structure for enterprise-ready web applications compared to a FastAPI + MongoDB stack.

3- Cloud-native deployments on Azure provide better performance and cost-efficiency than AWS for this type of web application.

4. Technology Stacks for Evaluation

Tier	Stack A	Stack B
Frontend	ReactJs	Python Flask + Jinja
Backend/API	Spring Boot (Java)	FastAPI (Python)
Database	PostgreSQL	MongoDB
Cloud Hosting	Azure	AWS or Google Cloud
Containerization	Docker	Docker
CI/CD	Azure DevOps	Jenkins

5. Proposed Features for Prototypes

- User registration & login
- Vocabulary lessons and categories

- Interactive quizzes (MCQ, matching)
- AI chatbot tutor integration (e.g., Claude or OpenAI)
- Audio-based pronunciation feedback
- User progress tracking

6. Evaluation Criteria

- **Performance:** Response time, stress test results, scalability.
- **Ease of Development:** Learning curve, tool compatibility.
- **Security:** Database security testing, user authentication.

7. Deployment Strategy

- Both stacks will be containerized using Docker.
- Stack A will be deployed on **Azure** (App Services or Container Instance).
- Stack B will be deployed on **AWS** (ECS or Lightsail).
- CI/CD pipelines will be implemented using Azure DevOps and Jenkins.

8. Deliverables

- GitHub repositories for both stacks
- Hosted prototype links
- Final report detailing:
 - Proof of hypotheses
 - Evaluation results of each stack
 - Recommendations for production

9. Timeline Overview

- Week 1–2: Finalize design & hypotheses, setup project structure
- Week 3–5: Develop and containerize both stacks
- Week 6: Deploy on respective clouds
- Week 7: Conduct testing and gather metrics
- Week 8: Write final report and submit all deliverables

10. Conclusion

This PoC will provide a solid foundation for building a scalable and effective language learning platform. By comparing stacks and testing AI features, the project will demonstrate the most viable architecture and tools for moving into full product development.