Sri Lanka Institute of Information Technology

Project Deployment Report

Project Title: AI-enabled-Intelligent-Assistant-to-Improve-Reading-and-Comprehension-Skills-in-English

Group Members (Group ID- RP24-25J-027)

IT21173790	Sooriyaarachchi .M.D.A
IT21158322	Senanayake W.G.B
IT21118340	Kumarathunga S.A.D.S
IT21182396	Ranaweera A.P

1.Project Overview

This report documents the development, deployment, and testing of our AI-powered English learning platform. It follows the Software Development Life Cycle (SDLC) and utilizes a microservices architecture to ensure modularity, scalability, and maintainability. The frontend is developed in React + Vite, while backend services are containerized and deployed on AWS ECS using CI/CD pipelines.

Key Metrics:

- Deployment Success Rate: 100%
- Downtime: 0 hours during deployment
- Uptime: 99.9% post-deployment
- CI/CD Status: Successful Docker image builds and GitHub Actions workflows

2. Deployment Objectives

Goals:

- Deliver a scalable, modular web application for English learning.
- Ensure each microservice operates independently and reliably.
- Deploy both frontend and backend components using modern DevOps practices.

Scope:

- Frontend built with React and Vite.
- 5 microservices for backend deployed on AWS ECS.
- Frontend deployed on Netlify.
- CI/CD with Docker and GitHub.

3. Software Specifications

3.1 Functional Requirements

- User registration and login
- AI-based vocabulary training
- Dynamic comprehension tests (basic and advanced)
- Speech error detection
- Real-time feedback and user progress tracking

- It should be able to analyze student-written input and predict their CEFR vocabulary proficiency level.
- It should use natural language processing to extract meaningful features like lexical richness and syntactic complexity.
- It should be able to generate vocabulary games (e.g., fill-in-the-blank, association games) based on user level and preferences.
- It should include a chatbot assistant capable of explaining word meanings,

3.2 Non-Functional Requirements

- Scalability via container orchestration
- High availability with auto-scaling ECS services
- Fast response via Vite + CDN (Netlify)
- Secure communication via HTTPS and JWT tokens

4. Pre-Deployment Activities

Environment Preparation:

- Set up AWS ECS clusters and services
- Created S3 bucket for assets
- Configured Netlify deployment from GitHub

Testing:

- Unit testing of each backend microservice
- End-to-end integration testing
- User Acceptance Testing (UAT) via Google Forms

Backup Plan:

- Codebase backed up via GitHub repositories
- AWS ECS configurations exported
- Daily snapshot backups for ECS volumes

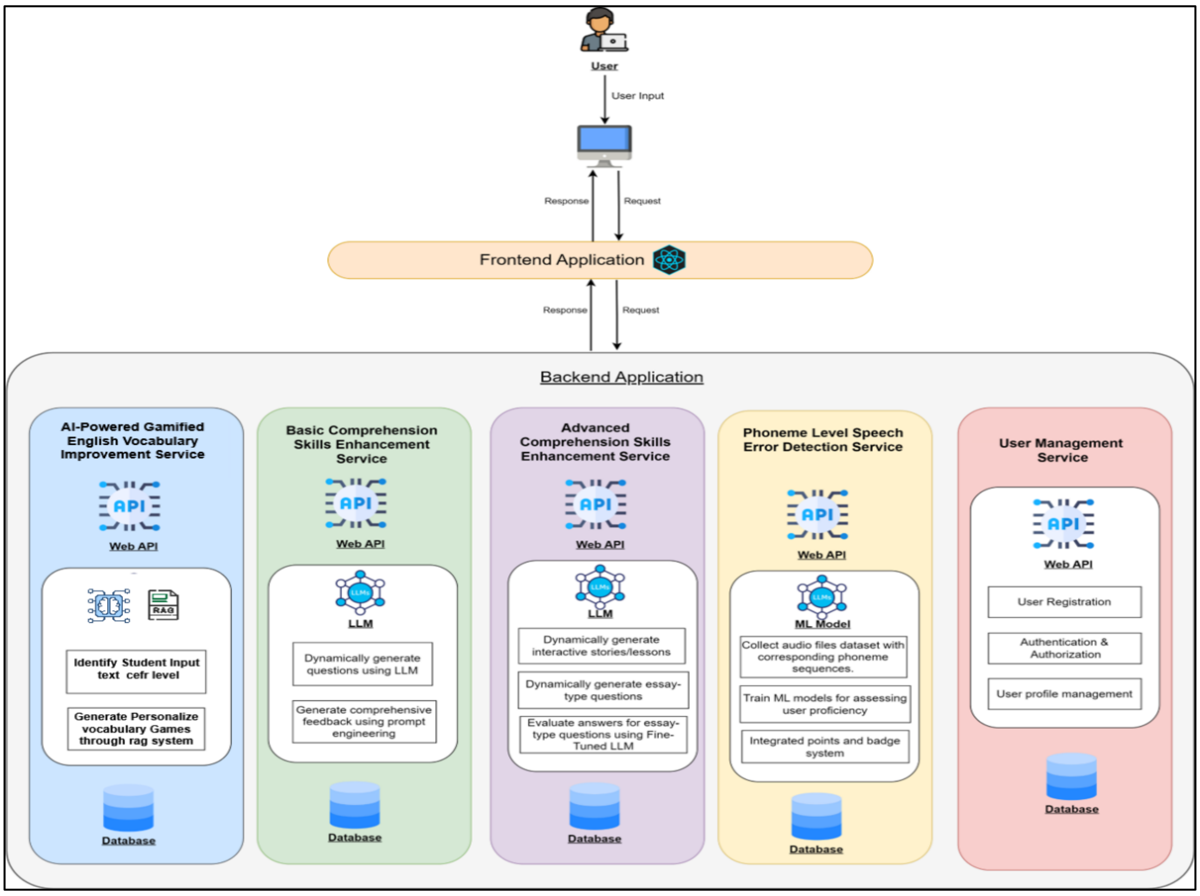
5. Architecture Overview

Based on your diagram, the system includes:

Frontend

- **React + Vite App** deployed on **Netlify**
- Handles user input and routes it to relevant backend services via API calls

Backend (5 Microservices)



User
User Input
Response
Request
Frontend Application
Response
Request
Backend Application
AI-Powered Gamified English Vocabulary Improvement Service
Web API
RAG
Identify Student Input text cefr level
Generate Personalize vocabulary Games through rag system
Database
Basic Comprehension Skills Enhancement Service
Web API
LLMs
LLM
Dynamically generate questions using LLM
Generate comprehensive feedback using prompt engineering
Database
Advanced Comprehension Skills Enhancement Service
Web API
LLMs
LLM
Dynamically generate interactive stories/lessons
Dynamically generate essay-type questions
Evaluate answers for essay-type questions using Fine-Tuned LLM
Database
Phoneme Level Speech Error Detection Service
Web API
LLMs
ML Model
Collect audio files dataset with corresponding phoneme sequences.
Train ML models for assessing user proficiency
Integrated points and badge system
Database
User Management Service
Web API
User Registration
Authentication & Authorization
User profile management
Database

Service Name	**Description**
AI-Powered Gamified Vocabulary Improvement	Uses AI to identify and gamify vocabulary gaps
Basic Comprehension Skills Enhancement	Dynamically generates questions using LLMs

Advanced Comprehension Skills Enhancement	Context-based dynamic comprehension, integrated with gamification
Phoneme-Level Speech Error Detection	Detects speech errors using ML and provides audio feedback
User Management Service	Handles user registration, login, and profile management

Each service:

- Exposes its own **REST API**
- Has its own **database**
- Is independently deployed via **Docker containers to AWS ECS (Fargate)**

6.Tools and Technologies

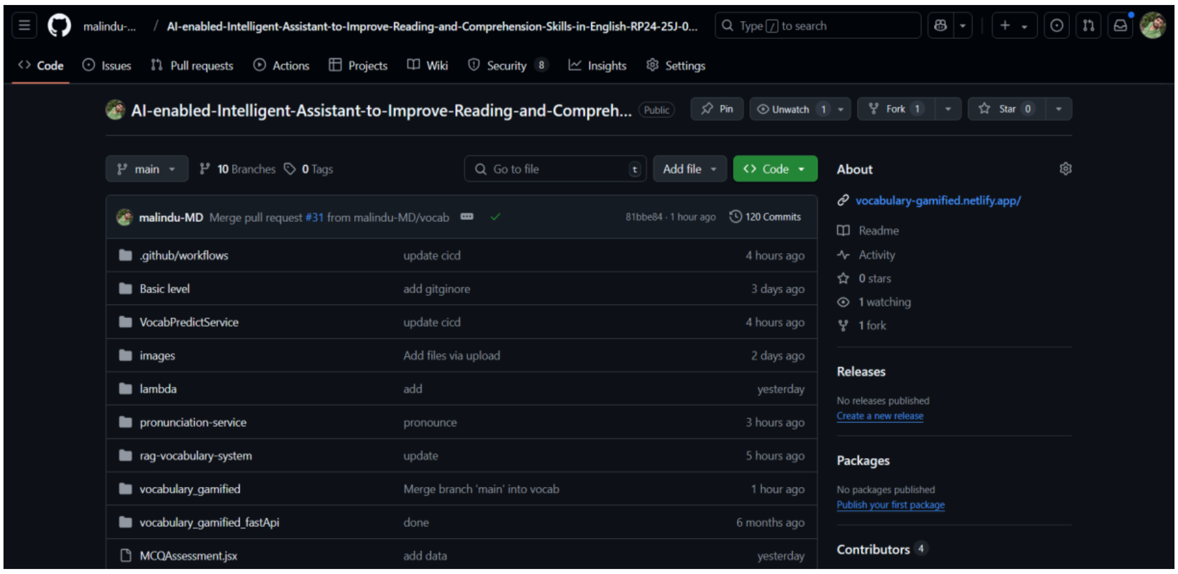
Category	**Tool / Platform**	**Version / Details**
Frontend	React + Vite	React 18 / Vite 5
Backend Services	Fast API	Service-specific stack
Database	PostgreSQL/Firebase /Pinecone Vector DB	Dedicated DB per service
Containerization	Docker / Docker Compose	Docker 24.0
CI/CD	GitHub Actions	Automated test/build/deploy
Hosting Platform	AWS ECS (Fargate)	Container orchestration
API Gateway	AWS API Gateway	Unified entry point for services
Static Hosting	Netlify	Frontend build and deploy

ML/AI Integration	Gemini AI / Speech Models/AWS /Higgin face/ Transformer Models / Lang Chain /NLP models/Embedding Models	Used For all Modules

7. Deployment Process

7.1 CI/CD via GitHub Actions

- Each microservice is in a GitHub repository



malindu-... / AI-enabled-Intelligent-Assistant-to-Improve-Reading-and-Comprehension-Skills-in-English-RP24-25J-0...
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security 8
Insights
Settings
AI-enabled-Intelligent-Assistant-to-Improve-Reading-and-Compreh... Public
Pin
Unwatch 1
Fork 1
Star 0
main
10 Branches
0 Tags
Go to file
Add file
Code

malindu-MD Merge pull request #31 from malindu-MD/vocab 81bbe84 · 1 hour ago 120 Commits

.github/workflows	update cicd	4 hours ago
Basic level	add gitginore	3 days ago
VocabPredictService	update cicd	4 hours ago
images	Add files via upload	2 days ago
lambda	add	yesterday
pronunciation-service	pronounce	3 hours ago
rag-vocabulary-system	update	5 hours ago
vocabulary_gamified	Merge branch 'main' into vocab	1 hour ago
vocabulary_gamified_fastApi	done	6 months ago
MCQAssessment.jsx	add data	yesterday

About

vocabulary-gamified.netlify.app/

Readme
Activity
0 stars
1 watching
1 fork

Releases

No releases published
Create a new release

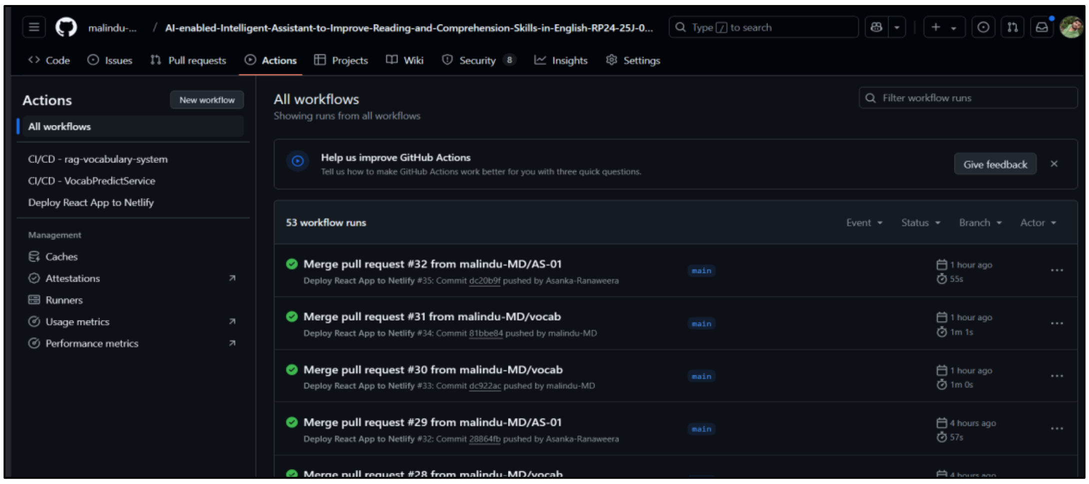
Packages

No packages published
Publish your first package

Contributors 4

Figure Error! No text of specified style in document.**.1 GitHub repository**

- CI triggers on push/pull_request:
  - Run tests
  - Build Docker image
  - Push to Docker Hub


malindu-... / AI-enabled-Intelligent-Assistant-to-Improve-Reading-and-Comprehension-Skills-in-English-RP24-25J-0...

All workflows
Showing runs from all workflows

CI/CD - rag-vocabulary-system
CI/CD - VocabPredictService
Deploy React App to Netlify

Help us improve GitHub Actions
Tell us how to make GitHub Actions work better for you with three quick questions.

53 workflow runs

Merge pull request #32 from malindu-MD/AS-01
Deploy React App to Netlify #35: Commit dc20b9f pushed by Asanka-Ranaweera — main — 1 hour ago — 55s

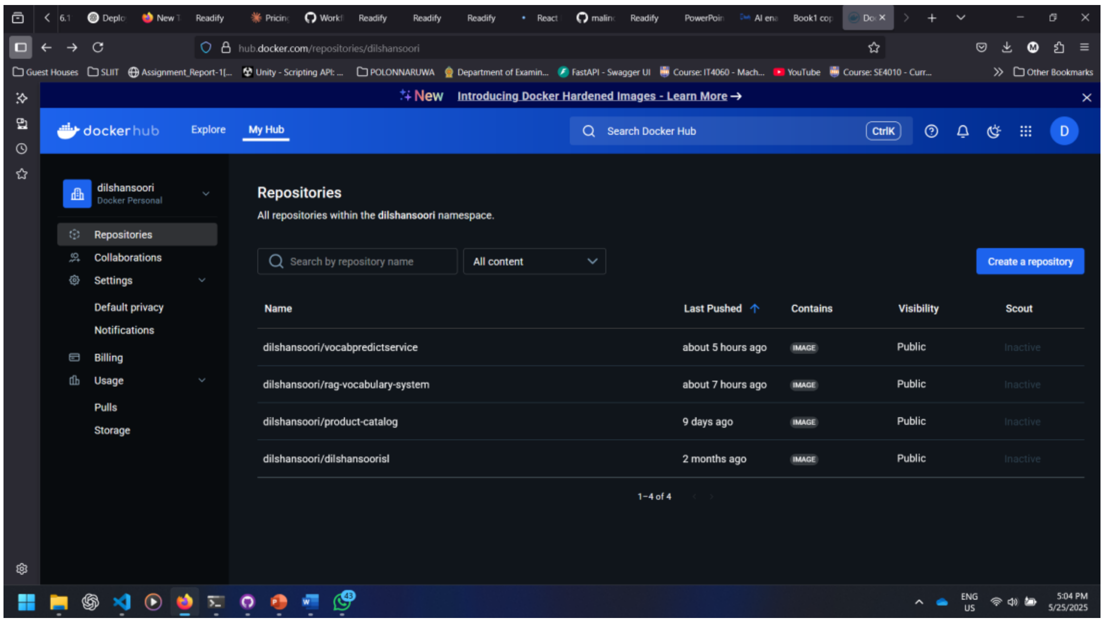
Merge pull request #31 from malindu-MD/vocab
Deploy React App to Netlify #34: Commit 81bbe84 pushed by malindu-MD — main — 1 hour ago — 1m 1s

Merge pull request #30 from malindu-MD/vocab
Deploy React App to Netlify #33: Commit dc922ac pushed by malindu-MD — main — 1 hour ago — 1m 0s

Merge pull request #29 from malindu-MD/AS-01
Deploy React App to Netlify #32: Commit 28864fb pushed by Asanka-Ranaweera — main — 4 hours ago — 57s

Merge pull request #28 from malindu-MD/vocab

Figure 2 GitHub Actions log showing successful build


hub.docker.com/repositories/dilshansoori

Repositories
All repositories within the **dilshansoori** namespace.

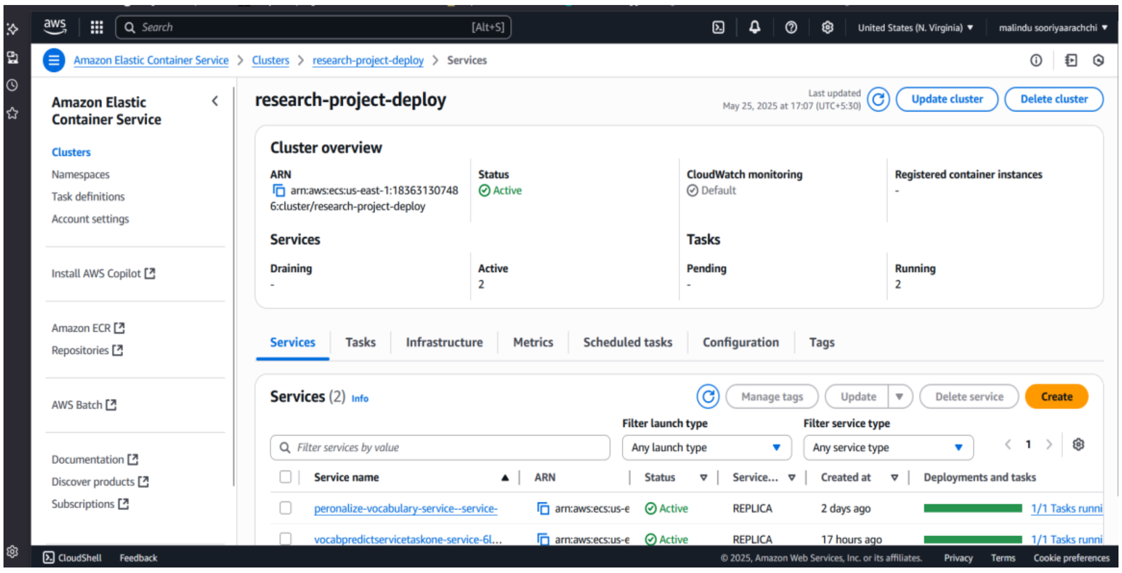
Name	Last Pushed	Contains	Visibility	Scout
dilshansoori/vocabpredictservice	about 5 hours ago	IMAGE	Public	Inactive
dilshansoori/rag-vocabulary-system	about 7 hours ago	IMAGE	Public	Inactive
dilshansoori/product-catalog	9 days ago	IMAGE	Public	Inactive
dilshansoori/dilshansoorisl	2 months ago	IMAGE	Public	Inactive

1–4 of 4

Figure 3 Docker HUB showing Docker images pushed

7.2 AWS ECS (Fargate) Deployment

- Services deployed independently to ECS


Amazon Elastic Container Service > Clusters > research-project-deploy > Services

research-project-deploy

Last updated May 25, 2025 at 17:07 (UTC+5:30)

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:us-east-1:183631307486:cluster/research-project-deploy	Active	Default	-

Services

Draining	Active
-	2

Tasks

Pending	Running
-	2

Services (2)

Filter launch type: Any launch type

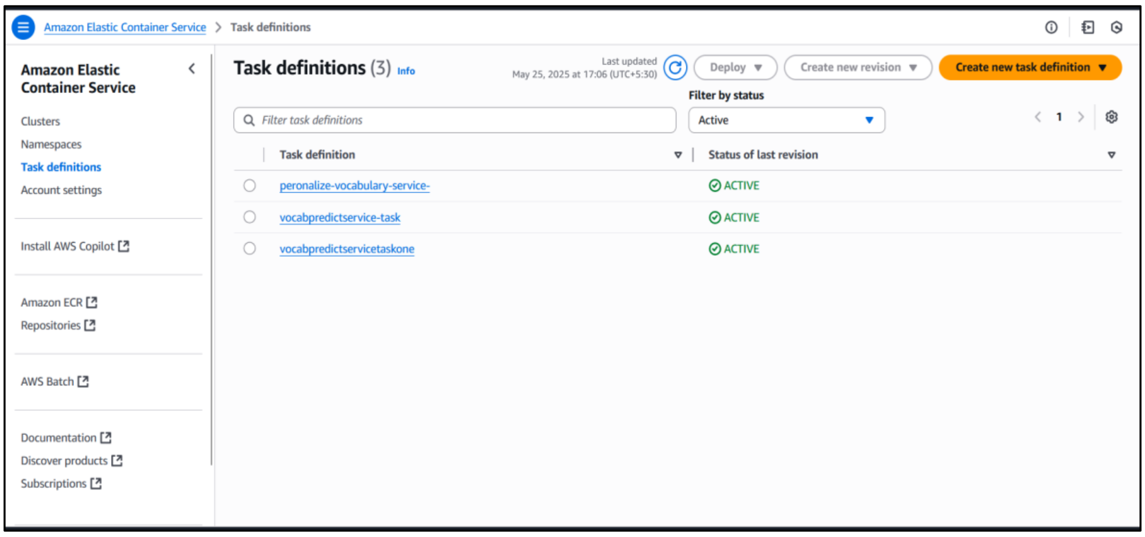
Filter service type: Any service type

Service name	ARN	Status	Service...	Created at	Deployments and tasks
peronalize-vocabulary-service--service-	arn:aws:ecs:us-e	Active	REPLICA	2 days ago	1/1 Tasks runni
vocabpredictservicetaskone-service-6l...	arn:aws:ecs:us-e	Active	REPLICA	17 hours ago	1/1 Tasks runni

© 2025, Amazon Web Services, Inc. or its affiliates.

Figure 4 AWS ECS dashboard with running services

- ECS Task Definitions used for auto-scaling


Amazon Elastic Container Service > Task definitions

Task definitions (3)

Last updated May 25, 2025 at 17:06 (UTC+5:30)

Filter by status: Active

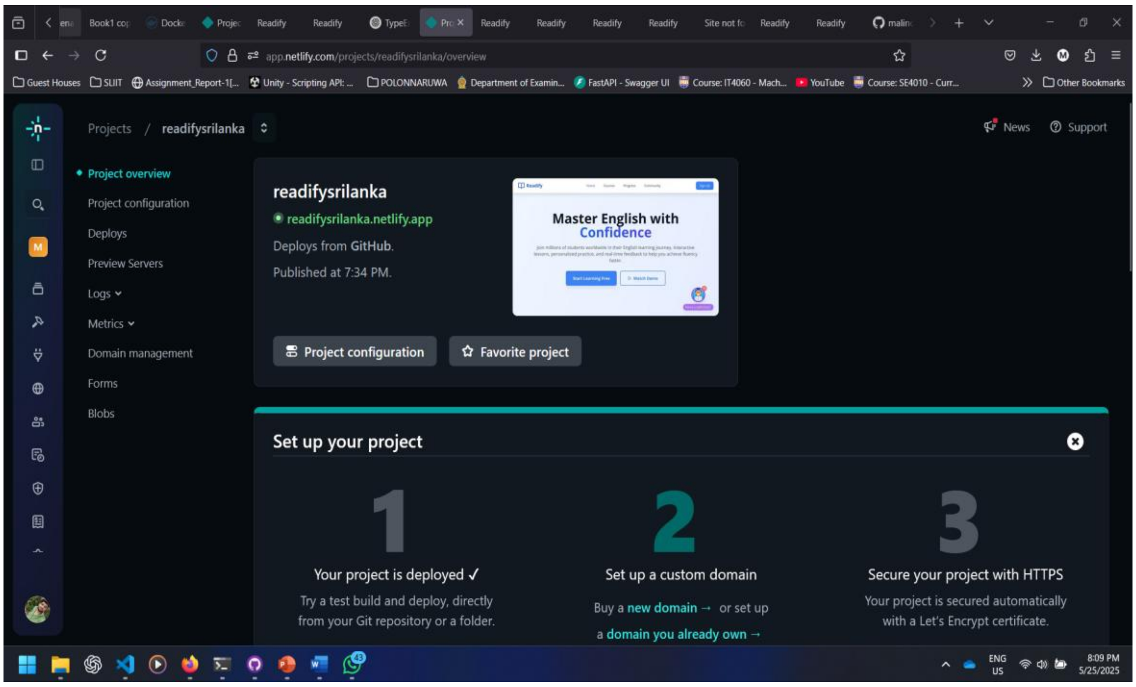
Task definition	Status of last revision
peronalize-vocabulary-service-	ACTIVE
vocabpredictservice-task	ACTIVE
vocabpredictservicetaskone	ACTIVE

Figure 5 AWS ECS Task Definition

- Load balancer + service discovery via AWS ALB
- Logs and metrics via CloudWatch

7.3 Frontend on Netlify

- Connected to GitHub repo
- Auto-deploy on main push
- Vite build generates production bundle
- Hosted globally with HTTPS



readifysrilanka
readifysrilanka.netlify.app
Deploys from GitHub.
Published at 7:34 PM.

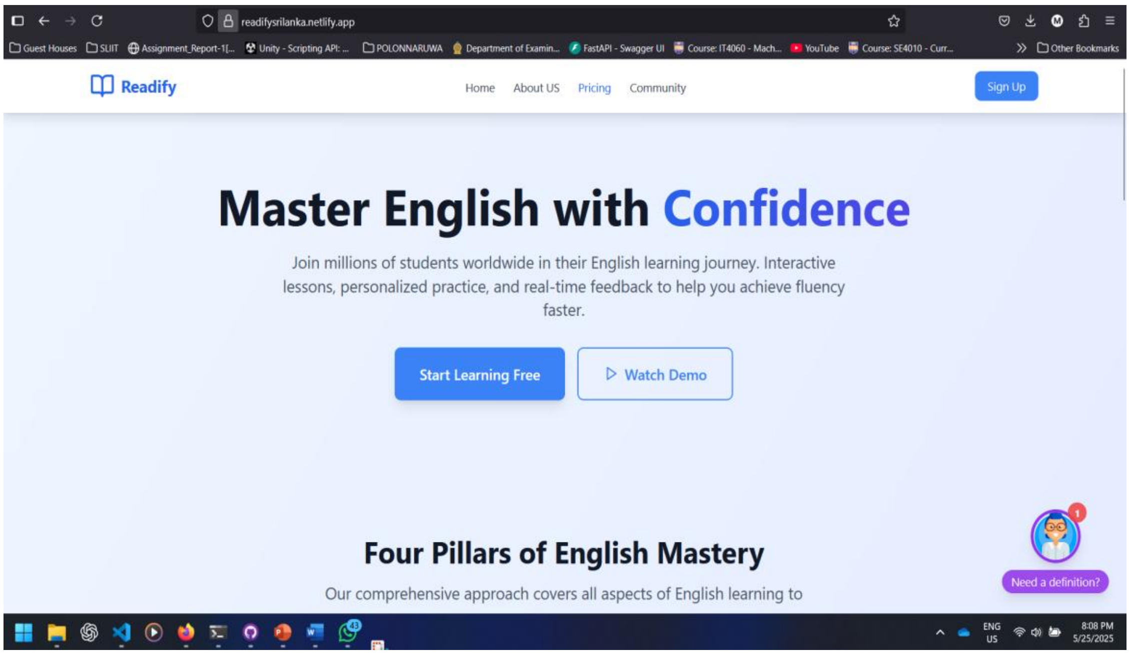
Set up your project

1
Your project is deployed ✓
Try a test build and deploy, directly from your Git repository or a folder.

2
Set up a custom domain
Buy a new domain → or set up a domain you already own →

3
Secure your project with HTTPS
Your project is secured automatically with a Let's Encrypt certificate.

Figure 6 Netlify Deployed Dashboard



Master English with Confidence

Join millions of students worldwide in their English learning journey. Interactive lessons, personalized practice, and real-time feedback to help you achieve fluency faster.

Four Pillars of English Mastery

Our comprehensive approach covers all aspects of English learning to

Figure 7 Frontend UI Preview

8. Cloud Configuration Details

Resource	Platform	Details

Frontend	Netlify	HTTPS, Auto-build on push
Backend Services	AWS ECS	Fargate tasks, 0.5 vCPU, 3GB RAM
Container Registry	Docker/Docker Hub	One repo per microservice
API Gateway	AWS Gateway	Routes API requests
Databases	PostgreSQL (RDS), Pinecone DB, Firebase	One per service, private subnet
CI/CD	GitHub Actions	Auto-build, test, deploy
Domain & DNS	Route 53 / Netlify	SSL, custom domain

9. Testing

9.3 Test Case Summary

Test ID	Component	Description	Status
TC01	User Login API	Login with correct credentials	pass
TC02	Speech Detection API	Upload audio, get feedback	pass
TC03	Vocab Service	Correctly worked	pass
TC04	Comprehension Generator	Generate questions from passage	pass
TC05	Frontend Rendering	User sees dashboard	pass

- Test Case for Vocab Service

Table **Error! No text of specified style in document.**:1 Test Case to CEFR Classification from Input Text

Test Case ID	**TC-01**
Scenario	Verify CEFR level prediction from input paragraph
Input	"The scientific community has debated the ethical use of AI."
Expected Output	CEFR Level: C1 or C2
Actual Output	CEFR Level: C1
Status	Pass

Table **Error! No text of specified style in document.**:2 Test Case to RAG Game Generation Based on Topic and Level

Test Case ID	**TC-02**
Scenario	Verify quiz generation using RAG system with input topic
Input	Topic: "Technology", CEFR: B1
Expected Output	Game JSON with MCQs and contextual vocabulary
Actual Output	10 MCQs returned with tech-related content
Status	Pass

Table **Error! No text of specified style in document.**:3 Test Case to Chatbot Vocabulary Response

Test Case ID	**TC-03**
Scenario	User queries chatbot for word explanation
Input	User message: "Define cryptocurrency"
Expected Output	Definition + example + related terms
Actual Output	Correct response generated
Status	Pass

Table **Error! No text of specified style in document.**:4 Test Case to Game UI Rendering

Test Case ID	**TC-04**
Scenario	Verify quiz renders correctly on UI
Input	Injected JSON with MCQ and options
Expected Output	1 question, 4 options, selection feedback

Actual Output	Accurate rendering and scoring
Status	Pass

Table **Error! No text of specified style in document.**:5 Test Case to Preference-Based Game Filtering

Test Case ID	**TC-05**
Scenario	Check game generation respects topic preference
Input	Preference: “Sports”, Level: A2
Expected Output	Questions about sports vocabulary
Actual Output	FITB and MCQs on “goalkeeper”, “match”, etc.
Status	Pass

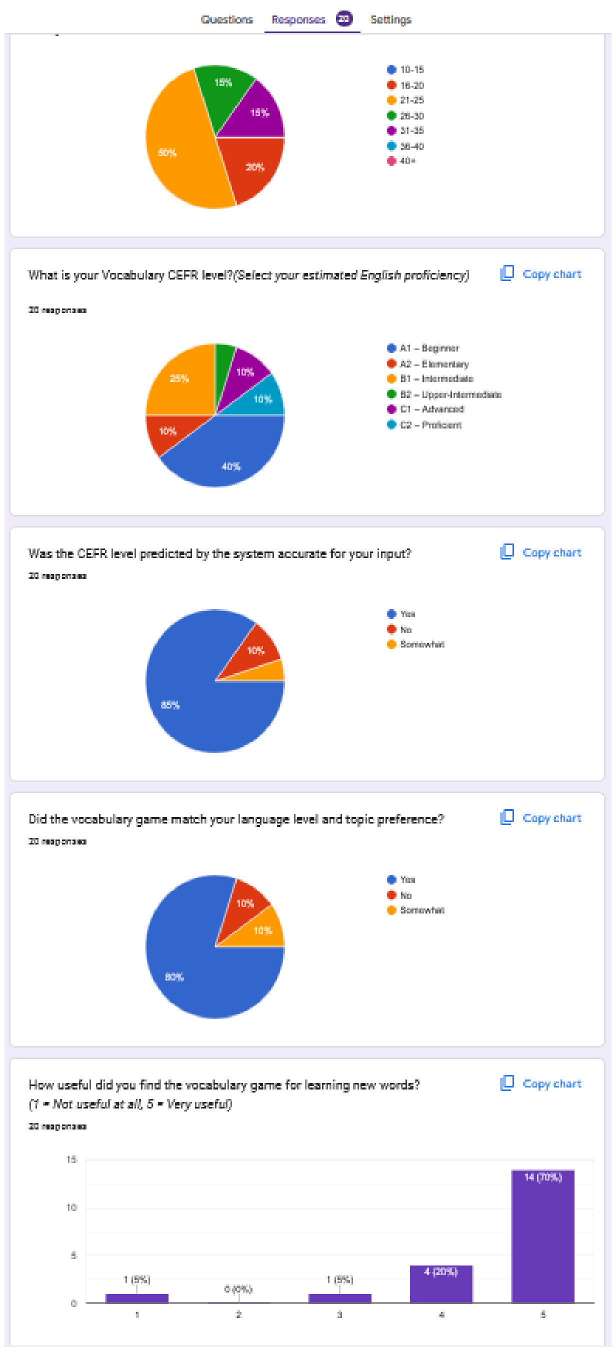
Table **Error! No text of specified style in document.**:6 Test Case to Fill-in-the-Blank Question Generation

Test Case ID	**TC-06**
Scenario	Check LLM generation of FITB questions
Input	Word: “firewall”, Level: B1
Expected Output	FITB JSON with 4 options
Actual Output	Game returned with structured JSON
Status	Pass

Table **Error! No text of specified style in document.**:7 Test Case to Hint Prompt Generation Accuracy

Test Case ID	**TC-07**
Scenario	Generate a hint for word without revealing it
Input	Word: “encryption”
Expected Output	Concise, indirect definition
Actual Output	“Used to protect information using codes.”
Status	Pass

- Google Forms for UAT to validate frontend usability, content relevance, and visual design with real user feedback



What is your Vocabulary CEFR level?*(Select your estimated English proficiency)*

20 responses

Was the CEFR level predicted by the system accurate for your input?

20 responses

Did the vocabulary game match your language level and topic preference?

20 responses

How useful did you find the vocabulary game for learning new words?
(1 = Not useful at all, 5 = Very useful)

20 responses

10. Issues and Resolutions

Encountered Issues:

- Netlify failed due to environment variable mismatch
- One ECS task failed due to port conflict

Resolutions:

- Updated Netlify env configuration

- Adjusted ECS task definition port settings

Impact Analysis:

- No downtime experienced by users
- Recovery time: < 30 minutes

11. Conclusion

Our system is a robust, AI-enhanced English learning platform built on microservices principles. It supports secure, scalable, and independent deployments of features and uses modern DevOps and cloud tools for reliability. Each component is tested and monitored, ensuring smooth performance across services.

GitHub Repositories: link

Demo Link: https://readifysrilanka.netlify.app/