Text

Description automatically generated with medium confidenceLingoLizard Design Document

**Name:** Michael Cullen

**Student ID:** C00261635

**Supervisor:** Dr Jamal Tauseef

**Date:** 4/12/2024

Contents

[Introduction 3](#_Toc184245469)

[Document Purpose 3](#_Toc184245470)

[Technologies 4](#_Toc184245471)

[Python 4](#_Toc184245472)

[Rasa 4](#_Toc184245473)

[PostgreSQL 4](#_Toc184245474)

[Django 4](#_Toc184245475)

[Git 4](#_Toc184245476)

[Postman 4](#_Toc184245477)

# Introduction

LingoLizard seeks to fulfil the growing need for effective language education by employing innovative real-time conversational learning techniques. The application focuses on fixing common issues with spelling and grammar and providing feedback that enhances the learning experience by integrating interactive practice, LingoLizard ensures that users not only learn the rules of language but also apply them in real-world scenarios.

LingoLizard aims to create an engaging and effective language learning experience that empowers users to improve their communication skills confidently and competently.

# Document Purpose

This document will outline LingoLizard's internal structure. It will describe the internal components and website and how they will function as a final product.

I will include a list of technologies I am using and how they are planned to be used, along with details of the alpha released for the first iteration.

# Technologies

## Python

A general-purpose programming language. Python is the foundation of LingoLizard, for backend logic, Rasa integration and database interaction.

## Rasa

An open-source framework for building AI-powered conversational chatbots. Rasa does the natural language understanding (NLU) and dialogue management in LingoLizard, so language learners get real-time grammar correction and conversational practice.

## PostgreSQL

A powerful open-source relational database system. PostgreSQL is used to store and manage structured data in LingoLizard, like user profiles, progress and scenario configurations.

## Django

A high-level Python web framework that makes building robust and scalable web applications easy. Django handles the backend of LingoLizard, API creation, data processing and front-end integration.

## Git

A distributed version control system that tracks changes in the code during development. Git is used to manage the codebase, collaborate with team members and ensure the project integrity across iterations.

## Postman

A popular API testing tool that makes sending requests and debugging API endpoints easy. Postman is used during LingoLizard development to test the REST APIs for backend communication with the front end and Rasa.

# Alpha release

## Language Selection

The user can choose the language they wish to practice, which determines the language in which all interactions with the bot will occur. The selected language dynamically influences the bot's responses, intents, and scenarios.

## Proficiency Selection

This option allows the user to specify their level of proficiency in the selected language from beginner, intermediate and advanced. This selection adjusts the difficulty of the scenarios, tailoring the prompts and challenges to suit the user’s skill level for an effective learning experience.

## Scenario Selection

Allows the user to choose a real-life situation they want to practice. Scenarios include Taxi, Restaurant, Hotel Check-In, and Asking for Directions. The selected scenario provides a focused context for language practice, making the learning experience immersive and practical.

## Django Interface

Displays a real-time chat window where users can interact with the bot. The chat interface shows the user’s messages and the bot’s responses.

# Flow Chart

A diagram of a computer program

Description automatically generated