



IsiPython: A Localised IsiXhosa Programming Environment for Inclusive Computing Education

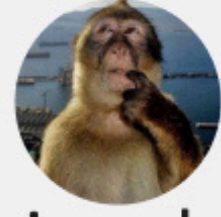
Team: Lutho Mngqibisa, Buhle Mlandu
mnglut008@myuct.ac.za, mlnhon001@myuct.ac.za,

Supervisor: Aslam Safla
asafla@cs.uct.ac.za

1 Background & Objectives

Programming in South Africa is mostly taught in English, creating linguistic and cognitive barriers for isiXhosa-speaking learners. IsiPython addresses this by localising Python's syntax, interface, and error messages into isiXhosa, making coding more accessible and culturally relevant. The project aims to reduce language-related challenges, enhance comprehension and confidence, and bridge isiXhosa learners to standard Python while promoting inclusivity in computing education. By enabling students to learn in their mother tongue, IsiPython helps them grasp complex programming concepts more easily and fosters a stronger connection between language, logic, and learning.

2 What is IsiPython ?



IsiPython is a web-based programming platform that brings coding to isiXhosa speakers. It allows learners to write programs using isiXhosa keywords such as ukuba (if) and chaza (define), view live translation to Python, and receive feedback or error messages in isiXhosa.

The platform also includes quizzes, coding challenges, bilingual documentation, and progress tracking through an interactive dashboard.

4 IsiPython Code

```
1 numbers = list(range(10000, 0, -1))
2 ngokulandelelana i ngaphakathi range(len(numbers)):
3     ngokulandelelana j ngaphakathi range(len(numbers) - 1):
4         ukuba numbers[j] > numbers[j + 1]:
5             temp = numbers[j]
6             numbers[j] = numbers[j + 1]
7             numbers[j + 1] = temp
8 print("Sorted!")
```

IsiPython keeps Python's structure but uses isiXhosa keywords, letting students code in their own language. For example, ukuba means if and ngokulandelelana means for. The example shows a bubble-sort algorithm written in isiXhosa syntax, translated to Python by IsiPython for execution — making programming clear, familiar, and inclusive.

5 Results

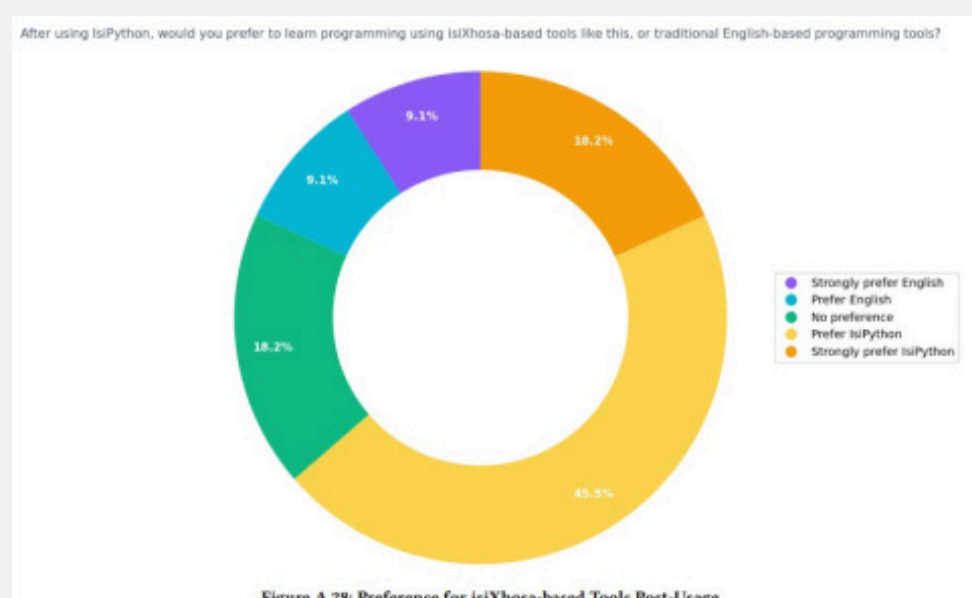


Figure A.26: Preference for isiXhosa-based Tools Post-Usage

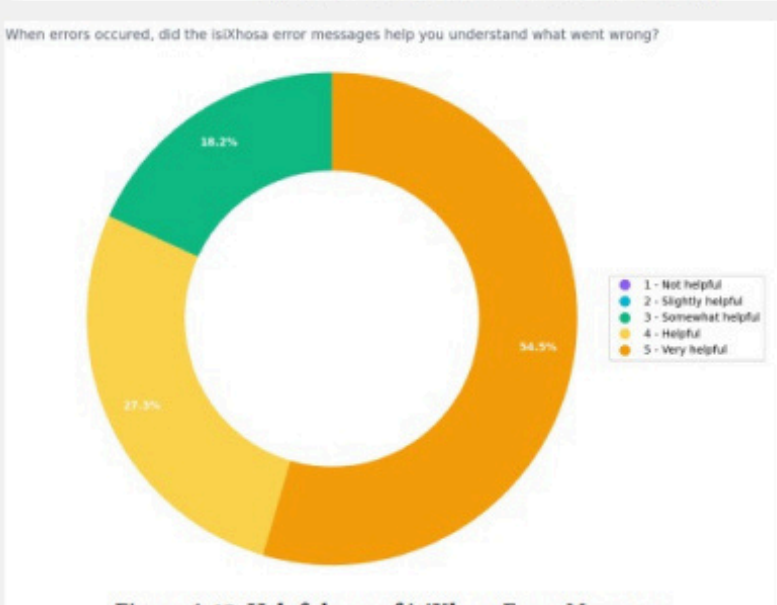


Figure A.19: Helpfulness of isiXhosa Error Messages

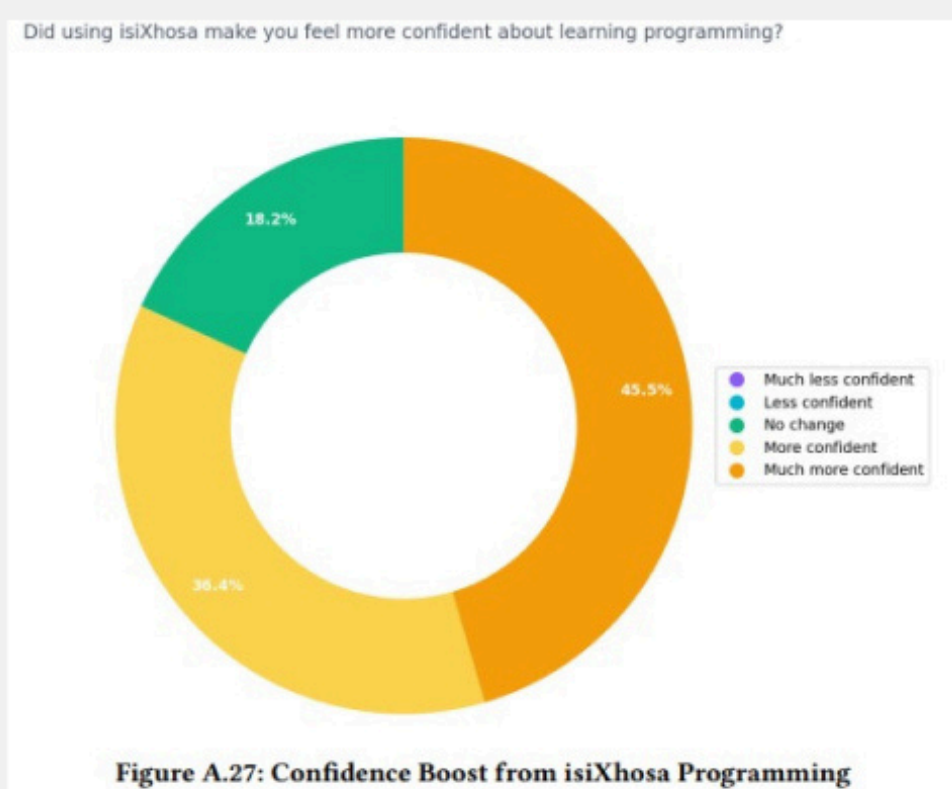
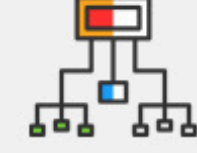
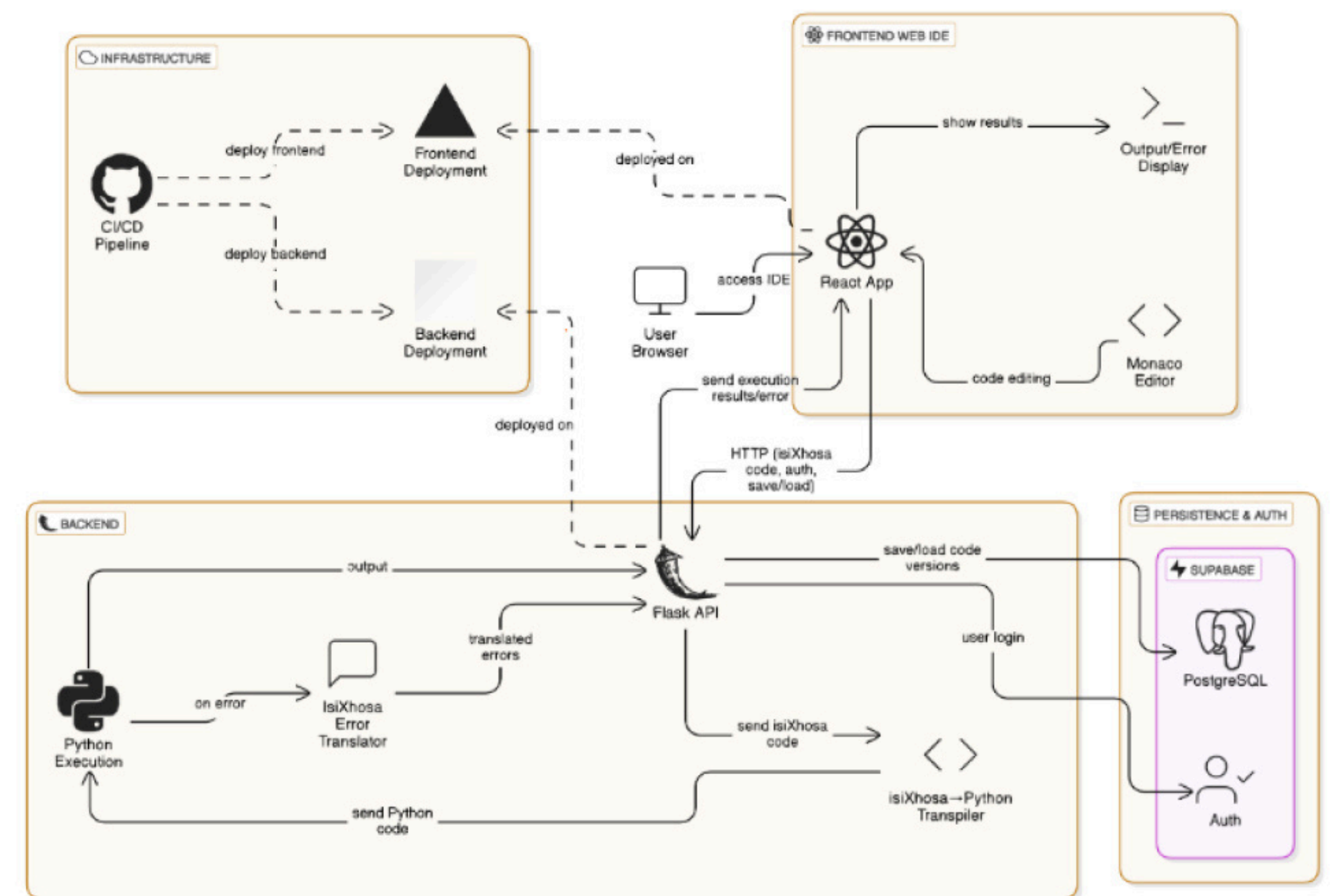


Figure A.27: Confidence Boost from isiXhosa Programming

System Architecture



IsiPython uses a modular, cloud-based design. The React IDE (with Monaco Editor) lets users code in isiXhosa and see live Python translation. The Flask backend converts isiXhosa code to Python, executes it, and returns bilingual output or errors. Supabase handles login and saves user progress, while CI/CD pipelines deploy updates to the frontend and backend.



Conclusion (Results Discussion)

Most participants (82%) found isiXhosa error messages helpful or very helpful in understanding what went wrong, highlighting the effectiveness of localised feedback in supporting debugging and comprehension. Over 80% of students reported feeling more confident learning programming when using isiXhosa syntax, suggesting that mother-tongue programming reduces anxiety and improves self-efficacy. Furthermore, 64% expressed a clear preference for isiXhosa-based tools over English ones after using IsiPython, indicating genuine acceptance and potential for adoption in teaching contexts.

Overall, the findings show that linguistic localisation not only enhances understanding but also strengthens learner motivation and inclusivity, demonstrating IsiPython's role as a viable model for decolonised, accessible computing education.



University of Cape Town

Department of Computer Science
www.sit.uct.ac.za

Authors

L. Mngqibisa
mnglut008myuct.ac.za.za
B. Mlandu
mlnhon001@myuct.ac.za

Supervisor

Aslam Safla
safla@cs.uct.ac.za

