Breaking the Language Barriers of Programming: An All-Inclusive and Personalizable Programming Platform — BASH-A

VIJAYAGANESH[1], SHUBHAM NANDWANA[2], RAVI KUMAR[3]
[1]CTO & Co-Founder, [2]CMO & Co-Founder, [3]CTO & Co-Founder

Abstract: There has been an increase in the growth of programming languages and software development tools, and this has been seen to favour the people who know English. This system has created a huge barrier for non-English speakers. The objective of this research paper is to propose the development of multi-language, customizable programming language software, BASH-A. Basically, BASH-A is going to ease the stages of learning by first allowing native-languages usage in coding, with a setting that can be self-done according to taste. In other words, it democratizes programming and makes it user-friendly. It creates an easy environment for beginners and greatly reduces the load associated with learning new syntax, further creating high engagements and increasing the retention of new programmers. It will also discuss the need for such a tool, its likely impact on the technology world, and the expected benefits for individual learners and professional programmers.

Keywords, BASH-A, Programming Education, Language Barriers, Multilanguage Support, Customization, User-Friendly Interface, Inclusive Programming, Beginner-Friendly, Syntax Customization, Tech Diversity, AI-Driven Personalization, Voice-Based Coding, Collaborative, Coding Environments

INTRODUCTION

The programming world is governed by the English language, making it a major hindrance to non-native English speakers who wish to learn and become proficient coders. Not only new learners but this language barrier has also inculcated potential diversity into the technology world. To this issue, we introduce BASH-A, a programming platform that enables users to code in their native language and further customize the syntax to suit their preferences. In this paper, we provide an overview of the concept, its customization features, and the potential benefits of having such a platform for both beginners and professional programmers.

Mission and Vision

Mission: Democratize programming through a language-agnostic platform for programming that will enable people from different linguistic backgrounds to learn to code and master it in their own mother tongue. The mission is to break the language barrier in technology and make programming easier to use, more accessible, and more inclusive for all beginners across the world.

Vision: We envision a global community of diverse and creative programmers contributing to the tech industry without language barriers. We envision being the number one platform for inclusivity in programming education so that technological advancement or innovation takes place through a greater, diversified, and more competent population of programmers.

LITERATURE REVIEW

The literature published regarding this topic uncovers problems non-native speakers of English face in the process of learning to program. These ideas have been supported by studies that have shown clear, major issues of linguistic barriers both in understanding and retention of programming. Several attempts at localizing programming languages have been reported, although most of these approaches fail to provide flexibility and personalization with respect to different preferences. Our work builds on this insight by offering a generalized platform, BASH-A, language-agnostic, but further allowing users to define their syntax, hence bridging the gap and making programming easier and more intuitive.

Current Research in the Market

The present research on the market indicates increased interest in tools on programming education across different linguistic backgrounds. While there may exist a plethora of localised programming languages, they lack the kind of personalization for the individual learner. Large companies like Google, Microsoft, and IBM have invested heavily in language translation services; however, there is definitely a requirement for integration with the programming environments. BASH-A would bridge this gap by bringing advanced translation and transcription technologies with a user-friendly interface for coding.

METHODOLOGY

BASH-A will be developed following an iterative design and development process. Our methodology includes the following main components:

Needs Analysis: Questionnaires and interviews to potential users to get information on problems they are facing and their personal preferences.

Design: User-friendly environment; multi-lingual support; custom syntaxes.

Testing: Testing for usability with users for feedback and iteration as necessary.

Deploy: Test-run it and track usage to know where more improvements could be made. Key Features Easy to Learn: It has simplified syntax and intuitive design, thus is accessible for beginners. User-Friendly: Clean, interactive interface for better learning. Multilanguage Support: One can code in native languages, which bridges the linguistic gap. Customization: Users could define their own syntax, hence making the programming environment of one's taste.

Pre-Built Keywords: A library of predefined keywords in various languages, which enables quickness at work.

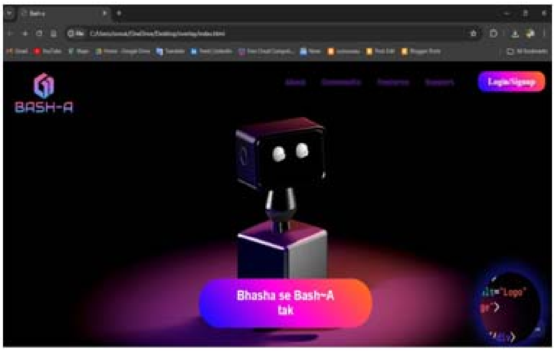
Tools and Technologies Used

A host of state-of-the-art tools and technologies have been used in the development and deployment of BASH-A. Each and every constituent has been chosen to fulfill the objectives of a robust, scalable, and user-friendly platform.

Following is an overview of all the details about the tools and technologies used:

Frontend Technologies

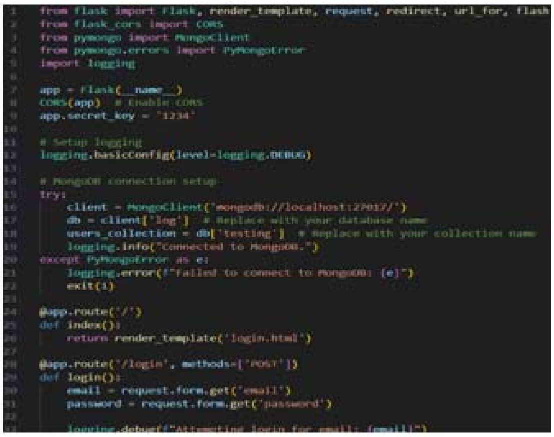
HTML— Hypertext Markup Language: This was used to structure the content on the web pages of the platform



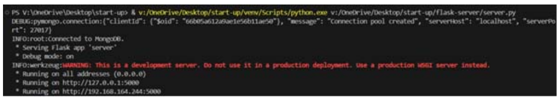
CSS (Cascading Style Sheets): This is used for styling purposes, to design web pages, in order to achieve a user-friendly and attractive interface. JavaScript: Makes a web application interactive and dynamic, thus more responsive and engaging to the user.

Backend Technologies

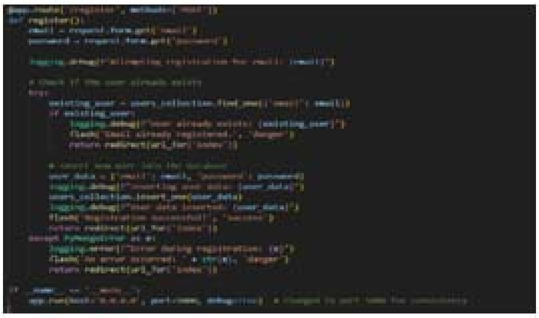
Python: A versatile programming language that can be used in handling server-side logic, handling requests, and management of the core functionalities of the platform.



```
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_cors import CORS
from pymongo import MongoClient
from pymongo.errors import PyMongoError
import logging

app = Flask(__name__)
CORS(app)  # Enable CORS
app.secret_key = '1234'

# Setup logging
logging.basicConfig(level=logging.DEBUG)

# MongoDB connection setup
try:
    client = MongoClient('mongodb://localhost:27017/')
    db = client['log']  # Replace with your database name
    users_collection = db['testing']  # Replace with your collection name
    logging.info("Connected to MongoDB.")
except PyMongoError as e:
    logging.error(f"Failed to connect to MongoDB: {e}")
    exit(1)

@app.route('/')
def index():
    return render_template('login.html')

@app.route('/login', methods=['POST'])
def login():
    email = request.form.get('email')
    password = request.form.get('password')

    logging.debug(f"Attempting login for email: {email}")
```



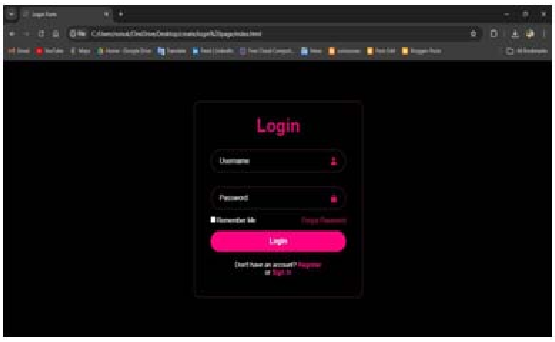
```
PS V:\OneDrive\Desktop\start-up> & v:/OneDrive/Desktop/start-up/venv/Scripts/python.exe v:/OneDrive/Desktop/start-up/flask-server/server.py
DEBUG:pymongo.connection:{"clientId": {"$oid": "66b05a612a9ae1e56b11ae50"}, "message": "Connection pool created", "serverHost": "localhost", "serverPo
rt": 27017}
INFO:root:Connected to MongoDB.
 * Serving Flask app 'server'
 * Debug mode: on
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.164.244:5000
```



```
@app.route('/register', methods=['POST'])
def register():
    email = request.form.get('email')
    password = request.form.get('password')

    logging.debug(f"Attempting registration for email: {email}")

    # Check if the user already exists
    try:
        existing_user = users_collection.find_one({'email': email})
        if existing_user:
            logging.debug(f"User already exists: {existing_user}")
            flash('Email already registered.', 'danger')
            return redirect(url_for('index'))

        # Insert new user into the database
        user_data = {'email': email, 'password': password}
        logging.debug(f"Inserting user data: {user_data}")
        users_collection.insert_one(user_data)
        logging.debug(f"User data inserted: {user_data}")
        flash('Registration successful!', 'success')
        return redirect(url_for('index'))
    except PyMongoError as e:
        logging.error(f"Error during registration: {e}")
        flash('An error occurred: ' + str(e), 'danger')
        return redirect(url_for('index'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)  # Changed to port 5000 for consistency
```

Database

MongoDB: A NoSQL database applied to all user data, syntax settings, and other information that needs to be stored in a flexible, document-oriented format.



3D Graphics

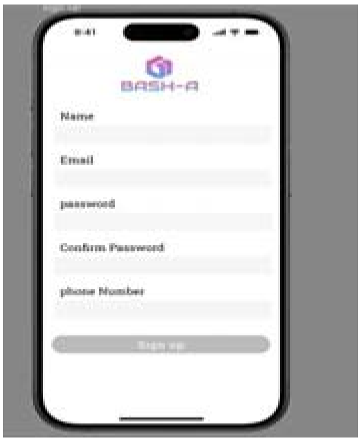
Spline: Applied in creating interactive 3D elements and visualizations that will increase the visual appeal and user experience across the platform.

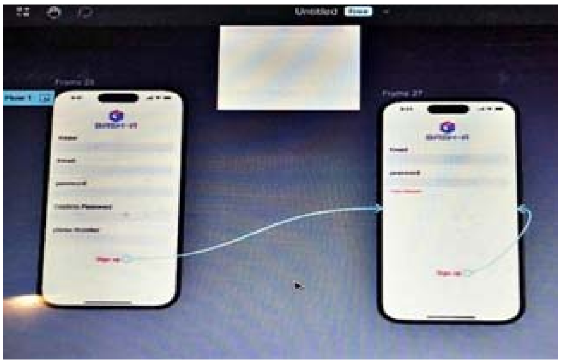
Cross-Platform Development

Dart: The language that is used together with Flutter to develop the frontend of the application.

Design and Prototyping

Figma: A web-based design tool for creating and collaborating on design prototypes and wireframes.





Canva: A graphics design platform; its application range spans into visual content such as promotional materials, user guide creation. Mobile Development

Android Studio: Integrated Development Environment, IDE for Android app development used in building the mobile version of the BASH-A platform.

Cloud Services

AWS EC2: Elastic Compute Cloud is a part of Amazon Web Services that provides computing capacity scalable in the cloud to host back-end services and manage large-scale operations.

AWS Translate Service: Applied in real-time language translation that allows users to code in their native language.

AWS Transcribe Service: Speech-to-text, providing the ability for voice-based coding and input.

Artificial Intelligence and Machine Learning

AI-ML: Supports intelligent algorithms to optimize the experience of the user by giving personalized suggestions and adaptability features of learning.

OpenAI: Provides high-order linguistic models and AI which shall be used in Natural Language Processing and other AI-driven capabilities.

Version Control

Git is a version control system designed for source code management or tracking changes to the source code, making it easier for people to collaborate in the development team.

Language Model

Gemini LLM Model: A large language model used to power several natural language comprehension tasks on the platform for more accurate translation and transcription.

CHALLENGES AND DIFFICULTIES

Technical Complexity: To make a sturdy system which cat-translates and personalizes into multiple languages without affecting the performance can be very challenging.

User Adoption: Major effort and benefits need to be shown to shift users from established programming languages and environments to a new platform.

Standardization: To create uniformity in functionality and performance across various customizations and languages.

Resource Allocation: It is an important factor that ensures sufficient resources are constantly spent on developing, testing, and maintaining the platform to respond to user feedback and further improvement.

Case Example of BASH-A Implementation in a Non-English-Speaking Country

Background

To demonstrate the effectiveness of BASH-A, we conducted a case study in a region where English is not the most spoken language. In this case, it was a rural area in India, and the language predominantly used was Hindi. Our participants were those high school students who had never done any programming before.

Methodology

Introduction to BASH-A: Introduction to the platform of BASH-A and initial training regarding the use of the interface and customisation of syntax.

Learning Modules: A series of programming lessons were conducted using BASH-A wherein the students coded in Hindi.

Customisation: The students could customize the syntax as per their comfort. For example, using "DIKHAO" for "PRINT".

Feedback Collection: Feedback was taken from the students regarding their experience on the platform, ease of learning, and challenges faced.

Results

Engagement: The students were very much engaged and interested in learning to code in their native language.

Ease of Learning: The intuitive interface and the ability to customize syntax drastically reduced the learning curve.

Feedback: The user-friendly environment and flexibility of syntax customization were highly acclaimed. Some technical difficulties, mainly with respect to translation accuracy and adaptation to different dialects, were brought out. Conclusion The case study proved that BASH-A was able to break the barriers related to linguistic aspects and give programming to non-English speaking people. The personalization ability of the platform was greatly appreciated; therefore, BASH-A has a bright future in different linguistic contexts.

Future Trends

AI-Powered Personalization: Artificial intelligence will be applied to offer learners personalized learning experiences and adjust the setting of the platform to individual learning styles.

Improved multilingual support: Development of support for more regional dialects and less commonly spoken languages.

Voice-based coding: Using speech recognition technologies to allow users to code in their native language by voice command.

Collaborative coding environments: Features that allow real-time collaboration among users of different linguistic backgrounds.

Cross-platform compatibility: Making sure that the platform is available on various devices, whether desktop, tablet, or mobile phone.

RESULTS

The implementation of BASH-A in the chosen region where English is not spoken returned very promising results:

High Engagement: The participants showed a highly enthusiastic and engaging nature towards learning programming with the use of BASH-A. Being able to code in their native language made the learning process more relatable and less daunting.

Ease of Learning: This intuitive interface, along with the ability to customize syntax according to individual preference, did much for lowering the learning curve. Participants testified that they easily grasped programming concepts and syntax.

The feedback from participants was quite encouraging. They enjoyed the user-friendly environment and flexibility in syntax customisation. Using familiar terms instead of traditional programming keywords allowed comfort and ease.

Technology Challenges: The main technical challenges were i) achieving an accurate translation and ii) dialect adaptation. These were overcome with iterative refinement and user feedback.

Better Retention and Higher Performance: The participants retained the programming concepts better and performed the coding tasks at a higher rate than in a traditional English-based programming environment.

CONCLUSION

This research paper thus adds a somewhat new paradigm to the art of making the exacting field of programming more accessible and user-friendly for any non-English speaking population by razing the linguistic barriers. Bash-A, being a highly customizable programming platform, could democratize programming education and empower people from linguistically diverse backgrounds. We envision that it would work with a global community of innovative programmers and make them contribute to the technological aspects without worrying about the hassle of a language barrier. If BASH-A succeeds, then, directly or indirectly, it will revolutionize programming education and help further inclusivity and diversity within the technology industry.