

Social Work Field Report on the Coding Curriculum

Student Name: Md. Sayedul Islam (2204007)

Course: HSS 2418

Institution: Dhaka University of Engineering & Technology, Gazipur

Date: January 13, 2025

Supervisor: Mir Md. Fazle Rabbi, Assistant Professor, Dept. of HSS, DUET

1. Introduction

This project aimed to create a Python coding course for high school students, focusing on polytechnic students at DUET Admission. We chose Python because it is part of their syllabus. Our goal was to teach them basic coding skills and spark their interest in technology and problem-solving.

We held online classes with hands-on activities to connect what they learned with real-life uses. The project also aimed to prepare students for the growing need for coding skills. We made the lessons engaging with visual aids, live demos, and exercises to help them understand and remember the topics. This project also helped polytechnic students prepare for the DUET admission test.

2. Project Objectives

The project aimed to:

- Create a beginner-friendly Python curriculum for Polytechnic Admission student.
- Improve logical thinking and problem-solving with coding exercises.
- Show how coding is useful in real life.
- Get feedback to improve the curriculum.
- Manage diverse student groups effectively.
- Introduce collaborative problem-solving through group projects.
- Assess and improve teaching methods regularly.
- Prepare students for academic and career opportunities in Python.

3. Why Polytechnic Students?

Polytechnic students were chosen because Python is part of their syllabus, making it relevant for their studies and DUET Admission exams. Learning Python gives them an advantage in exams and prepares them for advanced courses in engineering and technology.

4. Program Benefits

The project provided several benefits:

- **Practical Knowledge:** Students gained hands-on Python experience.
- **Career Preparation:** Coding projects made students ready for academic and job opportunities in tech.
- **Skill Development:** Students improved their logical thinking, teamwork, and project management.
- **Increased Motivation:** Interactive teaching methods and success stories inspired further learning.

5. Activities Performed

A. Curriculum Design

The curriculum covered:

- Programming basics (variables, data types, operators).
- Control structures (loops, conditionals).
- Functions and modular programming.
- Data structures (lists, dictionaries).

We used flowcharts and diagrams to explain programming logic. Periodic assessments checked student progress.

B. Student Management

We managed students by:

- Grouping them based on skill levels.
- Assigning leaders to facilitate peer learning.
- Using coding games and competitions to keep students engaged.
- Balancing individual and group activities with a detailed schedule.
- Creating an online platform for discussions and resource sharing.

C. Feedback Collection

We collected feedback to improve sessions. Students liked the hands-on approach but suggested more advanced examples like web development. Online format posed challenges like internet issues and less interaction.

6. Challenges Faced

A. Varied Skill Levels

Students had different coding skills. We provided extra resources and

one-on-one guidance for beginners and tailored exercises for all levels.

B. Resource Constraints

Some students faced internet and device issues. We used cloud-based platforms like Replit and recorded sessions to help them.

C. Maintaining Engagement

We used interactive activities and real-world examples to keep students interested. Quizzes and coding puzzles made learning fun.

D. Managing Student Collaboration

Different abilities in groups were challenging. Regular monitoring and reflection sessions helped balance group dynamics.

E. Balancing Curriculum and Time

We prioritized core concepts and used supplementary materials for self-paced learning. Iterative approaches reinforced difficult topics.

7. Learning Outcomes

- How to explain complex topics simply.
- How to design an effective beginner-friendly Python curriculum.
- The importance of continuous improvement through feedback.
- Strategies for managing diverse student groups.
- Flexibility in handling unexpected challenges.
- How to use technology to enhance learning.

8. Conclusion

Creating a Python curriculum was a rewarding experience. It introduced students to coding and problem-solving. The positive feedback confirmed the success of our hands-on teaching approach. Managing and motivating students taught me valuable lessons in classroom management.

I plan to include advanced topics like web development and data visualization in the future. Continuous feedback will help keep the program relevant and impactful.

This initiative will continue to inspire students and prepare them for tech careers.

9. References:

In this Project full code Resource:

[https://github.com/saidul-07/Academic/tree/main/Advanced%20Programming%20Sessional\(Python\)](https://github.com/saidul-07/Academic/tree/main/Advanced%20Programming%20Sessional(Python))

