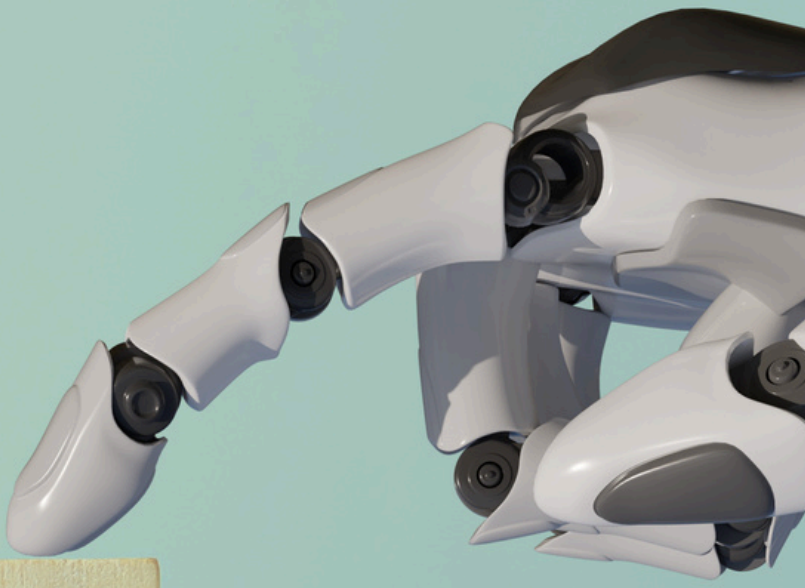
UNIVERSITI TEKNOLOGI MARA

What's what FSKM

EISSN: 2756-7729



UNBOXING FEELINGS:
MEASURING IPHONE HAPPINESS THROUGH SENTIMENT

INNOVATION COMPETITIONS:
GROWING CREATIVE MINDS FROM PRIMARY SCHOOL TO UNIVERSITY

LEARNING BY DOING:
ROBOTICS VERSUS MICRO:BIT FOR STEM EDUCATION

THE FUN SIDE OF MATH:
MAGIC TRICKS AND PUZZLES

COLLABORATING FOR SMARTER MOSQUES

How Chess Sharpens Your Mind and Math Skills

BEYOND THE CLAIM OF RANDOM SAMPLING

OCTOBER 2025/VOLUME 2

MAKING CODING FUN: WHY I DESIGNED Code-Ninja++ FOR MY STUDENTS

Tuan Norhafizah Tuan Zakaria
Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM),
Cawangan Negeri Sembilan, Kampus Kuala Pilah, 72000, Negeri Sembilan Darul Khusus, Malaysia.

tuannorhafizah@uitm.edu.my

THE STRUGGLE IS REAL: WHEN NON-IT STUDENTS FACE C++

As a lecturer, I've seen firsthand how many non-IT students struggle when it comes to learning C++ programming. For them, coding feels like stepping into a completely foreign world - full of strange symbols, strict rules, and logic that doesn't always make sense at first glance. Some students lose confidence quickly, and revision outside of class often becomes passive or even avoided altogether.

I wanted to change that. I wanted to build something that could make C++ more approachable, engaging, and even fun. That's how Code-Ninja++ was born.

BUILDING Code-Ninja++

Code-Ninja++ is a Progressive Web Application (PWA), which means students don't need to download or install anything. They simply open it in their browser and start learning. On the technical side, I used Python (Flask) for the backend, HTML and CSS for the frontend, and MySQL for database management. But for my students, what matters most is that it's easy to access on any device.

I designed the app as a one-stop hub for post-class learning. It includes short video tutorials, quizzes that appear immediately after videos, carry mark tracking, past year papers, and gamified elements like leaderboards and badges. I also integrated it with tools my students already use—Padlet, UiTM UFuture, and Quizizz—so they don't need to jump between multiple platforms. Figure 1 depicts the snapshot of Python, HTML, and MySQL that was used to develop Code-Ninja++.

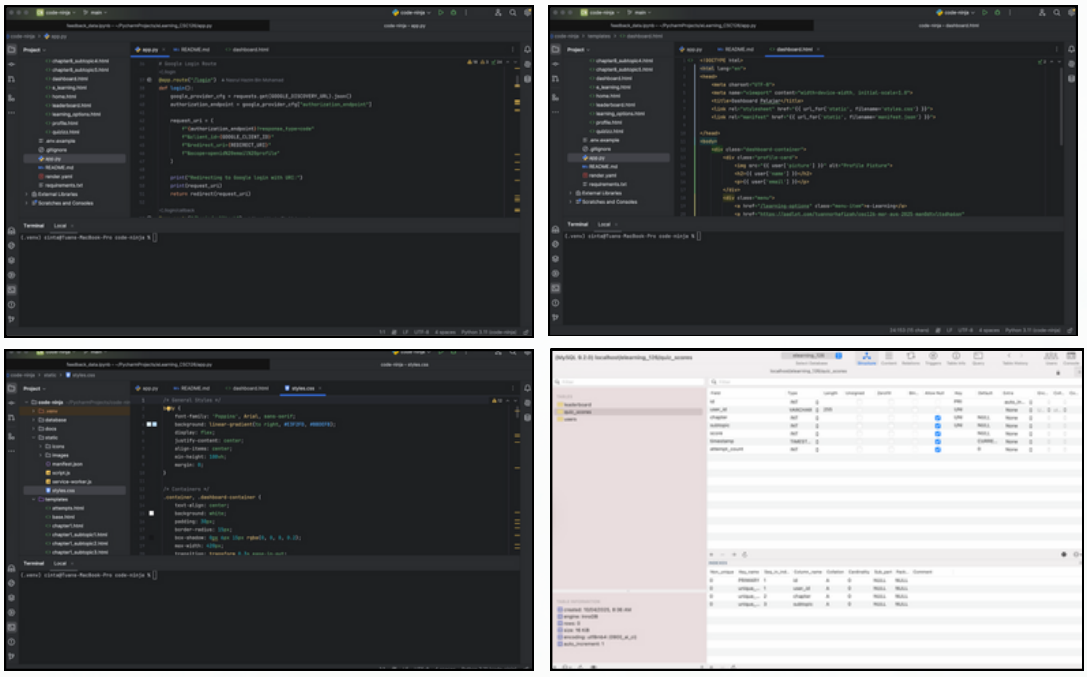


Figure 1: Python, HTML, CSS, and MySQL for Code-Ninja++ Development

LEARNING THROUGH PLAY

One thing I've learned as an educator is that students respond better when learning feels like play. That's why Code-Ninja++ uses gamification. Quizzes allow up to three attempts with instant feedback, leaderboards create healthy competition, and badges give students a sense of achievement.

Learning C++ is no longer just about memorising syntax—it's about building confidence step by step and tracking progress in real time. Figure 2 shows the main interfaces of Code-Ninja++.

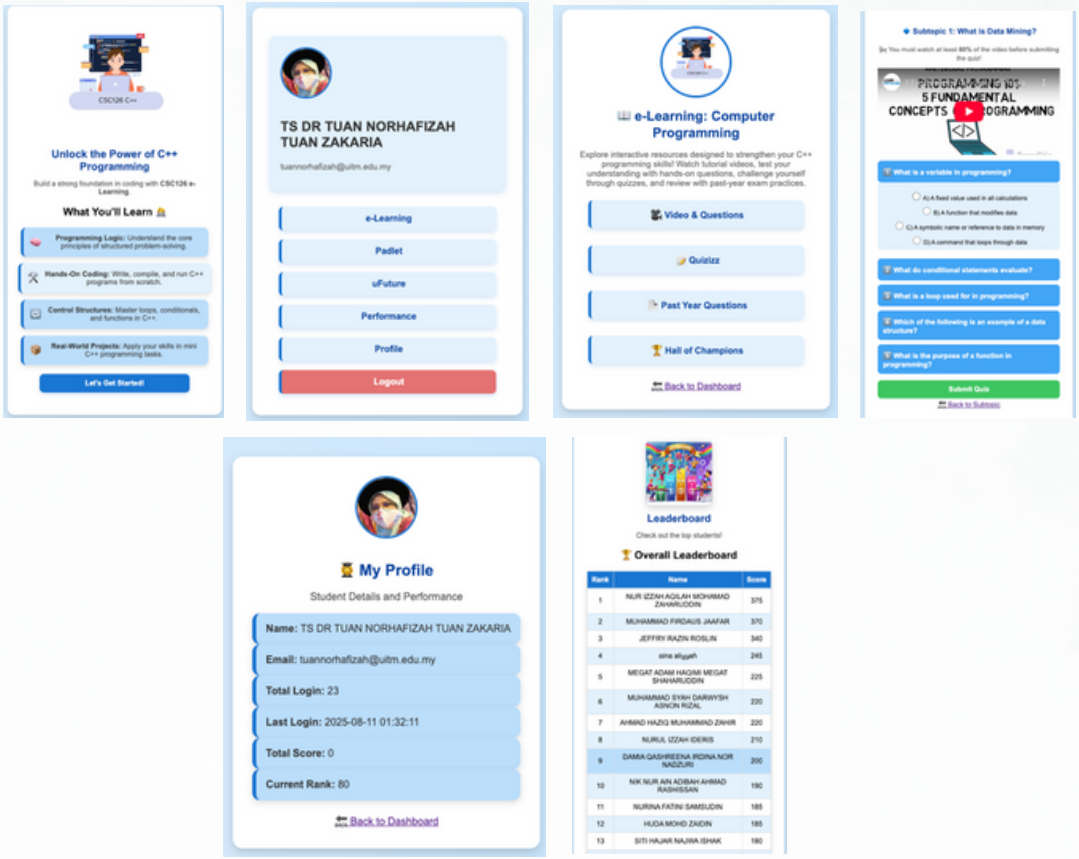


Figure 2: Main Interfaces of Code-Ninja++

WHAT MY STUDENTS THINK

The best part of this journey has been hearing directly from my students. In our feedback survey, most features received nearly 100% positive ratings. They loved the bite-sized videos, the instant quizzes, and the way the app brought all resources together in one place. The leaderboard and carry mark tracker, in particular, motivated them to keep practising.

Of course, a few pointed out areas for technical improvement, but overall, the message was clear: Code-Ninja++ works, and it makes learning C++ more enjoyable.

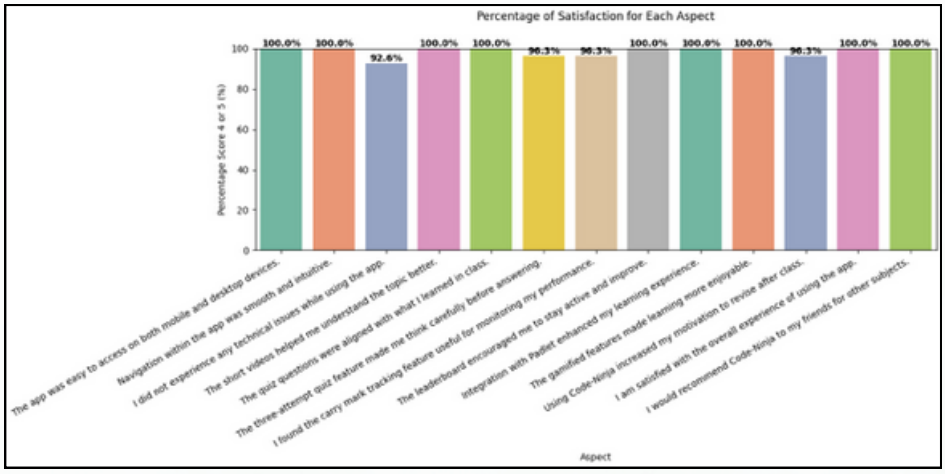


Figure 3: Student Satisfaction Ratings for CodeNinja++ Features

Based on our student survey in Figure 3, the results were very encouraging. As shown in the chart, most features received 100% satisfaction ratings, especially for accessibility, smooth navigation, and the alignment of short videos with classroom lessons. Students particularly valued the instant quizzes, leaderboard challenges, and the carry mark tracker, which made learning more interactive and fun.

The only slightly lower scores were for technical reliability (92.6%) and motivation from leaderboard/carry mark features (96.3%). These areas give us valuable feedback for future improvement. Overall, the analysis confirms that Code-Ninja++ is not just accessible—it truly enhances student engagement, confidence, and independent learning.

WHY THIS MATTERS

For non-IT students, learning C++ isn't just about coding. It's about developing logical thinking, problem-solving skills, and the confidence to engage with technology. With Code-Ninja++, I've seen students move from passive learners to curious, motivated individuals who are ready to tackle challenges.

LOOKING AHEAD

I see Code-Ninja++ as just the beginning. My next goal is to expand it to other programming languages like Python and Java, and to share it more widely with other faculties and universities. Education is evolving, and tools like this can help us keep pace with the way students learn best today—anytime, anywhere, and in ways that both engaging and rewarding.

For me, CodeNinja++ isn't just a project. It's a reflection of what I believe teaching should be: accessible, engaging, and empowering for every student.