Genetic Algorithms based adaptive e-learning framework

CS4099 Project Final Report

Submitted by

H Kailas Nath
Reg No: B210518CS

Serena Anthony
Reg No: B210031CS

Sreeshma Sangesh
Reg No: B210035CS

Under the Guidance of
Dr. Anu Mary Chacko

Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

May 06, 2025

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT, KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2025

CERTIFICATE

Certified that this is a bonafide record of the project work titled

GENETIC ALGORITHMS BASED ADAPTIVE E-LEARNING FRAMEWORK

done by

H Kailas Nath

Serena Anthony

Sreeshma Sangesh

of eighth semester B. Tech in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology Calicut

Project Guide	**Head of Department**
Dr. Anu Mary chacko	Dr. Subashini R
Associate professor	Associate Professor

DECLARATION

We hereby declare that the project titled, **Genetic Algorithms based adaptive e-learning framework**, is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : NIT Calicut
Date : 06-05-2025

Signature :
Name : H Kailas Nath
Reg. No. : B210518CS

Signature :
Name: Serena Anthony
Reg No: B210031CS

Signature :
Name: Sreeshma Sangesh
Reg No: B210035CS

Abstract

Classical E-learning portals make use of the One-Size-Fits-All approach when catering to diverse students. In the absence of an instructor, this method fails to motivate the students, in spite of excellent content. This project aims to create an adaptable e-learning framework that provides personalized learning paths and content for students based on their individual cognitive profiles using the Machine Learning approach. The framework is divided into four modules – User Profile Module, Instructional Content Management Module, Personalized Learning Path Generation Module and Evaluation Module. User Profile Module evaluates the student through tests and creates an individual profile identifying the student's learning style and cognitive ability. The Instructional Content Management Module is responsible for storing instruction content, identifying the nature of learning content, and managing the dependency graph that captures the prerequisite relationships to ensure pedagogical coherence. The Personalized Learning Path Generation Module combines the content of the above two modules. It creates an individualized learning path appropriate for each student and updates it dynamically as the student progresses through the learning path. The personalized path is generated using an ML model that leverages Genetic Algorithms (GA) in conjunction with Layered Topological Sort (LTS). The evaluation module includes a pre-assessment to measure prior knowledge, post-topic assessments to monitor progress, and a summative assessment after course completion to evaluate overall learning gains. The approach was evaluated by creating a prototype in the Moodle Platform that obtains the student profile through an initial assessment, evolves a personalized learning path and walks them through the generated path. For testing the prototype, a course based on Grade 6 CBSE NCERT Science chapters, *'Food: Where does it come from'* and *'Components of food'* was created. Suitable learning content was picked from online resources and classified on the basis of selected features. Students from nearby schools attempted the course and the results demonstrate the feasibility of the proposed approach.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have supported and guided us throughout the course of our final year project.

First and foremost, we extend our deepest thanks to our project guide, Dr. Anu Mary Chacko, for her invaluable guidance, constant support, and encouragement. Her insights and expertise have played a crucial role in shaping our project and helping us overcome challenges along the way.

We are also immensely grateful to our panel members Dr. A Lijiya and Dr. Vidhya Kamakshi for their constructive feedback and valuable suggestions that has significantly enhanced the quality of our work.

A special note of thanks goes to PhD scholar, Ms.Sheejamol P.T for her continuous support, thoughtful guidance, and the generous time she devoted throughout the course of our project.

And last but not the least, we would like to thank the administration, staff and students of Dayapuram Residential School and Spring Valley School for granting us the opportunity to conduct essential tests for our project. Their cooperation and support were vital to the successful implementation and real-world validation of our ideas.

This project would not have been possible without the collective support of all these individuals and institutions, and we remain truly thankful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Neurodiverse students often face significant challenges in traditional learning environments due to rigid, one-size-fits-all approaches that do not accommodate their unique cognitive and sensory needs. Traditional methods, which are based on text-based materials, lectures, and standardized tests, often overwhelm students with learning differences, causing difficulties in focus, comprehension, and social interaction. For example, students with Attention Deficit Hyperactivity Disorder (ADHD) might struggle to maintain attention during online lectures or assignments without interactive elements. Personalization in education, such as providing multimodal content, flexible pacing, and individualized learning pathways, can address these challenges by tailoring the experience to suit each student's strengths and needs.

E-learning platforms are especially well suited for this, as they can offer customizable environments where neurodiverse students can control their sensory input, receive immediate feedback, and access content in formats that match their learning preferences. The flexibility and adaptability of e-learning can provide an inclusive educational experience that empowers neurodiverse students. E-learning is useful for students in remote regions, who lack local tutors, as it can provide high-quality, customized education remotely. This makes e-learning an essential tool for ensuring accessible education, regardless of geographical limitations.

The field of personalization and e-learning for neurodiverse students has gained interest in the past few decades. Multiple research studies have been conducted in these domains. The initial research primarily involved the use of rule-based techniques, which limit the options offered to the users.

While some of the previous works have focused on screen readability, text-to-speech and navigation ease, creating content tailored to match the cognitive and pedagogic preferences of the students still remains a challenge.

As awareness of neurodiversity grew, especially in the early 2000s, educators and developers began considering the unique needs of neurodiverse students. Traditional education systems started adopting Individualized Education Plans (IEPs), and this idea slowly translated into e-learning environments, as platforms began to explore personalized learning approaches.

Recent attempts on personalization involve the use of ML, which proves to be a tool of utility in providing a dynamic learning environment. Decision trees have been used to classify students into categories, grouping similar students for content recommendations. Reinforcement learning (RL) along with its variations have been widely used in e-Learning, optimizing personalized learning path recommendations based on rewards like feedback and scores. Neural networks have been associated with personalizing learning and detecting disengagement during the learning process. GA is primarily engaged in evolving optimal learning paths and course content.

This work further explores the development of an adaptive e-learning framework for school students. The focus of this work is to explore the feasibility of identifying students' learning styles and aligning content accordingly, while generating learning pathways that maintain pedagogical rigor and respect the prescribed instructional order.

This thesis is organized into several chapters, each contributing to a comprehensive understanding of the problem and the solution approach.

The following chapter presents the literature survey conducted as part of the study. It provides a comprehensive review of existing work relevant to the problem domain, including current methodologies and techniques. It also highlights the limitations of previous approaches that motivated this research and identifies the specific research gaps that this project aims to address.

Chapter 2

Literature Survey

This study aims to develop an adaptive framework for E-learning, with particular emphasis on addressing the diverse needs of learners, including neurodiverse individuals. The literature survey primarily investigates current e-learning systems, examining how they are implemented and identifying existing limitations and gaps in effectively supporting varied learner profiles.

The salient findings from the reviewed papers are listed below and tabulated in Table. 2.1.

Eleanor et al [1] highlighted three main factors that affected neurodiverse students in the domain of online education: Firstly, cognitive factors such as attention and focus during study sessions and online communication; Secondly, instructional factors such as interacting with the e-learning platform; Thirdly, social factors such as stigma surrounding neurodiversity, lack of knowledge among tutors on educating neurodiverse students and the financial constraints involved with online education.

Sheila [2] suggested the use of Universal Design for Learning (UDL), which supported students who learn differently by providing students with multiple means to access each of these brain networks. UDL frameworks generally consider options that provide multiple means of engagement, representation, and action or expression. The UDL framework identified three aspects of information representation that can pose significant challenges to students: perception, language, symbols and comprehension.

Various papers on the usage of ML algorithms for e-learning were surveyed. The architecture of such frameworks along with their user profiling strategies was studied in detail.

Yiouli et al [3] suggested a robust architecture consisting of several key components, each with distinct functions. The User Model (Profiles) served to store all user-related data, including personal information, preferences, and performance metrics, enabling the system to deliver customized instruction based on individual learning styles and needs. The Domain Model (Content) was responsible for storing educational material, which is structured to accommodate various learning styles and preferences, ensuring that content is presented in a way that meets users' specific requirements. The User Interface utilized information from both the User Model and Domain Model to create a user-friendly presentation of the learning material, facilitating easy navigation and comprehension. The User Monitoring component tracked user behavior during interactions with the system, identifying interests and areas of difficulty, which informs adjustments to the User Model for ongoing personalization. Together, these components formed a cohesive architecture that enhances the adaptability and user-centric nature of the e-learning system, ultimately improving the learning experience for each individual user.

Mohammed and Yasser [4] proposed a general structure for an e-learning framework where the modules were broken down into lessons, and lessons further broken down into LOs. These LOs are tagged and stored in repositories for future retrieval. Students interacted with the Learning management system, which collected the learning preference of the students. An intelligent learning personalization module then renders a collection of the LOs to the students, depending on their learning styles which were previously determined.

Ilie et al [5] explored how AI and machine learning enhance e-learning by personalizing content, optimizing learning paths, and improving student outcomes. It highlighted benefits like increased engagement and performance, while also addressing challenges such as data privacy, system complexity, and integration issues.

The above paper majorly highlights the benefits of AI in education, thereby facilitating the shift towards ML-based implementations in our work to enable adaptive, personalized, and scalable learning solutions.

Samina et al [6] proposed a personalized e-learning framework based on Q-learning algorithm and Markov Decision Process (MDP). The algorithm develops a sequential path recommendation tailored to the user's learning preferences and past interactions. The proposed method combines dynamic programming, Q-learning, and MDP to create a robust framework for personalized adaptive learning, demonstrating significant improvements in learner engagement and performance. This work also highlighted the potential of adaptive sequencing in personalized learning.

Wafaa et al [7] introduced 'APPEAL', an adaptive personalized e-learning platform designed using a Deep Q-Network Reinforcement Learning (DQN-RL) model to create a dynamic learning model that detects and adjusts to individual learning styles and cognitive levels. It combines Visual, Aural, Read/Write, Kinesthetic (VARK) modalities with gamification to enhance engagement and learning effectiveness, addressing a gap in existing research. The students are presented with questions from each level of the Bloom's taxonomy, to evaluate their knowlege gain. The platform allows for real-time adaptation based on student interactions and incorporates a comprehensive feedback mechanism alongside content and exercise navigation.

Xiao-hong TAN ,Rui-min SHEN and Yan Wang [8] introduced a method called Personalized Course Generation and Evolution based on GA (PCE-GA). The initial course content was generated using an Automatic Course Generation System (ACGS), a teacher-driven system. The ACGS constructed courses based on a teaching outline and a domain structure provided by educators. GA helped in constructing a sequence of domain concepts and learning materials by evaluating the difficulty levels of the concepts and the time spent by the learners. The authors presented a novel algorithm that converges towards an optimal solution while considering multiple objectives. This algorithm leverages the stochastic nature of genetic algorithms, enhancing the effectiveness of course content selection. This paper introduces Partially Mapped (PMX) crossover, a specific crossover operation used in the GA to combine two parent chromosomes. It ensures that all genes are represented in the offspring without repetition. This paper highlighted the usage of Layered Topological Sort (LTS) to ensure that the generated path remains aligned with the instructional flow and topic prerequisites. A fitness function has been formulated in this work, which correlates the fitness of a chromosome to the difficulty level of the topics and the time taken to complete them. It further incorporates the variance observed across the genes to evaluate the overall fitness of an entire chromosome.

Mu-Jung Huang, Hwa-Shan Huang, and Mu-Yen Chen [9] talked about the importance of learning from mistakes and rooted their approach in the mastery learning theory, which asserts that students should achieve a high level of understanding in one topic before progressing to the next. The GA proposed by them constructed a learning path to strengthen the topics for which the learners give incorrect pre-test responses. They illustrated the use of a case-based reasoning approach (CBR) to develop a summative examination of the concepts. CBR helps the system select relevant questions and evaluation strategies based on similar past learner profiles and their outcomes, effectively making the evaluation process context-aware and learner-specific.

Oluwatoyin C. Agbonifo and Olanrewaju A. Obolo [10] highlighted the need to consider the relation degree that exists between various course concepts in order to deliver an optimal learning path to the student using the proposed GA. They made use of Inverse Document Frequency(IDF) to measure the degree of relationships that exists between the concepts.

As per the papers [9] and [10], traditionally e-learning platforms take into consideration the learner's preferences and interests but they failed to consider factors like the learner's ability, the difficulty of the material and the relationships that exist between topics.

Maycock [11] proposed an e-Learning platform that implemented GA to build a lesson.

He suggested the creation of learning objects(LO) which are stored in repositories. The LOs are analyzed by an automated content analyzer to extract relevant metadata. A profile is built for each user to obtain their cognitive abilities and pedagogic preferences. GA was run on the LOs, producing populations of lessons which are a combination of LOs. The author introduced Minimum Expected Learning Experience (MELE), the threshold fitness value for a course to be delivered to the student. To assess the learner's reading ability, the author incorporated the Flesch Reading Ease (FRE) metric, a well-established readability metric. Additionally, N-back test is introduced here as a tool to assess the working memory of the students. This multi-metric profiling enables the GA to select LOs that are not only pedagogically relevant but also cognitively accessible to the student.

To identify effective user profiling strategies, the following papers were surveyed, each offering insights into modeling learner characteristics for personalized education.

Inés et al [12] discussed various profiling strategies and grouping for the people interacting with the platform into categories like users, tutors and developers. They suggested the use of self declared profiles to obtain preferences directly from users to better suit their learning requirements. Other factors such as demographics, behavioral and performance data can be used to group users into categories with similar learning styles. They, however, highlighted the need to include more profiling attributes that consider feedback, emotions and the behaviors of students. The paper emphasizes on the need for profile, but self-declared profile may not be correct especially in the case of children. So, there is a need to automate the generation of profiles.

Paper [11] highlighted the use of the N-back test to measure working memory. The feasibility and validity of this test have been further explored in the literature. The following two papers discuss the theoretical basis and empirical evaluation of the N-back task in assessing working memory.

Susanne et al [13] discuss the N-back task and its application in measuring working memory (WM).The paper indicates that lower reliability estimates were observed in the 3-back versions, which might be due to increased error variance. This suggests that as the n value increases, the task becomes more challenging and may not yield consistent results for all participants. The study also mentions that the easy 1-back tasks might reflect ceiling effects, meaning that many participants could perform at a high level, making it difficult to distinguish between their abilities. This could imply that 1-back tasks might be too easy for some students, while 3-back tasks could be too difficult leading to flooring effects.

Pelegrina et al [14] found that age-related improvements in performance were more pronounced at the 2-back level compared to the 1-back level. This suggests that as children grow older, they are better able to handle the complexities of the 2-back task, making it a more effective measure of working memory development. The data suggests that 11-12 year old students are capable of handling the demands of the 2-back task effectively.

To further enhance user profiling, studies focusing on the grouping of students based on similar characteristics were examined. The following two papers propose clustering strategies and discuss their effectiveness and benefits in the context of personalized learning.

Efrati et al [15] highlighted how clustering effectively identifies diverse learning styles among students, allowing for tailored educational experiences that meet individual needs. It enabled the analysis of behavioral patterns, revealing similarities within student groups that can inform instructional strategies. The process could be automated to dynamically enrich student models over time, ensuring that learning profiles remain current and relevant. Mapping clusters to learning styles had shown encouraging results in their research, indicating that clustering can enhance educational outcomes by aligning teaching methods with learner characteristics.

Iftikhar et al [16] highlighted that clustering helps group students based on learning behavior, enabling personalized content delivery and improving learning outcomes. It discovered hidden patterns in large, diverse educational datasets and identifies outliers automatically. Their recommended method (CFSFDP-HD) adapted without needing prior knowledge of cluster numbers, making it more robust and efficient than traditional methods like K-Means or DBSCAN. This integration strengthens e-learning systems by making them intelligent and adaptive.

Genetic Algorithms (GAs) are particularly well-suited for the learning path generation problem, as they are highly effective in solving sequencing and combinatorial optimization problems—making them ideal for arranging learning objects in a personalized and pedagogically coherent order. While GA has shown promise in handling personalization and optimization tasks within e-learning systems, their application remains relatively underexplored. This perceived potential, coupled with limited prior implementations, motivated a more detailed survey of GA-based approaches to adaptive learning.

Paper [8] introduces a specialized crossover operator, Partially Mapped Crossover (PMX), where two crossover points define a matching segment that establishes gene mappings between parents. This approach preserves high-quality gene segments while ensuring gene uniqueness in offspring. The effectiveness of this method in maintaining diversity and transmitting beneficial traits is highlighted in the original paper, which is cited below.

Sivanandam et al [17] introduced the Partially Mapped Crossover (PMX) as a method used mainly for permutation-based problems like the Traveling Salesman Problem. In PMX, two parent solutions exchange a portion of their sequences between two crossover points. The mapping ensures that no duplicate elements occur, preserving the validity of the offspring.

This method maintains both the order and position information from the parents, making it suitable for problems where element uniqueness matters.

In addition to PMX, several other GA operators were explored during the literature review; one such operator that drew particular attention is introduced here.

George et al [18] introduced an operator in GA called Gene repair. The operator is composed of two tasks: fault detection and fault correction. They proved that the results obtained are either global optimal solutions or close to the optimal solutions. Further, the solutions appeared to be produced in smaller number of generations, indicating faster convergence of GA.

2.1 Discussion

Based on the literature review, the following points have been identified:

1. Neurodiverse students require a framework that takes into account their cognitive abilities such as memory capacity and information processing to provide a tailored learning environment. To enable this, appropriate profiling strategies must be employed to accurately capture individual learning needs and preferences.

2. Students usually find it difficult to sustain motivation to complete a course, often resulting in ineffective learning. It is therefore essential to provide a dynamic learning path which is reordered when the student hits a roadblock, thus ensuring continuous learning.

3. The framework should ensure that the path is coherent, taking into consideration the relationship between the topics in the course.

4. Grouping students (clustering) based on learning characteristics helps tailor content delivery, improves learning outcomes, and enables adaptive systems by revealing behavioral patterns and learning styles.

5. A feedback system in the form of assessments is necessary to evaluate student understanding and track their progress.

6. Gamification significantly improves student interaction with the framework, improves information retention and often provides better results.

In the survey of existing e-learning frameworks, it was found that dynamic content delivery often leverages Reinforcement Learning (RL) or Genetic Algorithms (GA).

Reference	Domain	Findings
[1]	Neurodiversity	Factors affecting neurodiverse students: Cognitive factors like attention and focus, instructional factors like interaction with the platform, and social factors such as stigma and lack of knowledge among tutors.
[2]	Neurodiversity	Universal Design for Learning.
[3]	Architecture	An architecture for e-learning framework consisting of 4 main components: User Model (Profiles), Domain Model (Content), User Interface, and User Monitoring.
[4]	Personalization in e-Learning	An outline for an e-learning framework: An intelligent learning personalization module that assembles learning objects (LOs) from a repository and delivers them to the user depending on their learning style.
[5]	ML in e-Learning	The use of various ML algorithms in e-learning for personalization and path generation.
[6]	Personalization in e-Learning	A sequential learning path recommendation system using Q-Learning and MDP, considering sequential behaviors and learning styles.
[7]	Personalization in e-Learning	An e-learning framework designed using DQN-RL, VARK, and Bloom's taxonomy.
[8]	Personalization in e-Learning	PCE-GA and LTS, a fitness function incorporating difficulty, time taken and gene-level fitness variance
[9]	Personalization in e-Learning	Construction of learning paths using GA, CBR, and mastery theory.
[10]	Personalization in e-Learning	Curriculum sequencing based on the relation degree that exists between various concepts, measured using IDF.
[11]	Personalization in e-Learning	Generating course content from a collection of LOs using GA, considering the cognitive and pedagogical preferences of the user (e.g., Flesch reading score for readability and N-back test for working memory).
[12]	User Profiling	Self-declared profile not sufficient to build a learning path; attributes accounting for user feedback, emotions, and behaviors must be included.
[13]	User profiling	N-back test and its application in measuring working memory.
[14]	User profiling	Effectiveness of 2-back test in measuring working memory, especially among 11-12 year old students
[15, 16]	User profiling	Effectiveness of clustering students in e-learning and personalization
[17]	Genetic Algorithms	Unique crossover operator to prevent repetition
[18]	Genetic Algorithms	Gene repair operator that detects faults in chromosomes in each generation and corrects the faults.

Table 2.1: Summary of Key Findings in Literature Review

- RL typically requires carefully crafted reward functions, which are challenging to design for neurodiverse learners due to the variability in success metrics. In contrast, GA offers greater flexibility in defining fitness functions, allowing optimization across multiple, sometimes conflicting, objectives.

- The crossover and mutation operations in GA introduce diversity into the learning pathways, preventing the system from becoming trapped in repetitive content loops, which can adversely impact engagement for neurodiverse learners. RL, on the other hand, may struggle to effectively explore such a broad solution space.

- Unlike RL, which often requires large volumes of interaction data to learn effective policies, GA can perform well even with limited data by leveraging prior knowledge embedded in the fitness function and initial population design. This makes GA more suitable for personalized learning environments with constrained user interaction data.

GA operates on a population of solutions, enabling simultaneous exploration of multiple learning paths. This population-based approach is particularly advantageous when addressing diverse learning needs, as it naturally accommodates variability without requiring extensive retraining. Thus, we have chosen to implement GA for evolving dynamic learning pathways.

2.2 Gap Analysis

From the literature review, the following gaps have been identified:

1. Many proposed frameworks rely on simplistic or self-declared learner profiles, which fail to capture key cognitive and behavioral traits. This limits the system's ability to deliver truly personalized learning experiences, especially for neurodiverse learners who require deeper profiling for effective adaptation.

2. Many systems optimize for a single metric without considering the trade-offs between multiple learning parameters such as difficulty, readability, and compatibility. This limited optimization reduces the level of personalization and fails to support the diverse learning needs of neurodiverse students.

3. Implementations based on Reinforcement Learning (RL) require meticulously designed reward functions and extensive interaction data—resources often lacking at the beginning of a course. These limitations hinder the system's adaptability and effectiveness. When applied to frameworks for neurodiverse learners, such constraints become even

more pronounced, as the variability in cognitive needs and success metrics demands a more flexible and responsive approach than RL typically affords.

4. Ensuring that a meaningful and pedagogically coherent path can be constructed from the given course content is a crucial constraint—one that is often overlooked by several existing frameworks.

5. While some frameworks generate initial paths, they lack mechanisms to dynamically adjust content based on ongoing learner performance (e.g., repeated failure or disengagement). This leads to static learning experiences that do not address evolving learner needs.

6. Several frameworks remain theoretical or are limited to isolated simulations. They are not validated through full-scale integration into established platforms like Moodle, making it difficult to assess their feasibility in practical educational settings.

2.3 Conclusion

In summary, while existing research has made meaningful contributions to adaptive e-learning, notable gaps remain—especially in areas like user profiling, pedagogical structuring, and optimization methods. These limitations underscore the need for a more holistic and adaptable framework, particularly suited for neurodiverse learners. Given the advantages previously discussed, Genetic Algorithms (GA) have been selected as the core machine learning technique for the proposed system.

The following chapter outlines the problem definition and sets forth the objectives that shape the design of this framework.

Chapter 3

Problem Definition

Develop an e-Learning framework tailored for neurodiverse students by using genetic algorithms to dynamically evolve personalized learning content and pathways. The framework should ensure that the learning path adheres to a given course dependency graph. The system should adapt to individual cognitive abilities and diverse learning styles, ensuring an inclusive and effective educational experience.

Chapter 4

Methodology

This section outlines the architecture of the proposed framework, with its key features grounded in insights drawn from the literature survey. The overall design is inspired by the approach discussed in paper [3], which emphasizes user profiling and domain modeling. Additionally, paper [4] informs the domain model through a structured repository of learning objects.

Building on these foundations, the proposed e-learning framework consists of four core components: User Profile, Instructional Content, Personalized Learning Path Generation, and Evaluation. Each component contributes critically to delivering a personalized learning experience and is described in detail in the following subsections. Fig. 4.1 shows the overview of the proposed architecture.

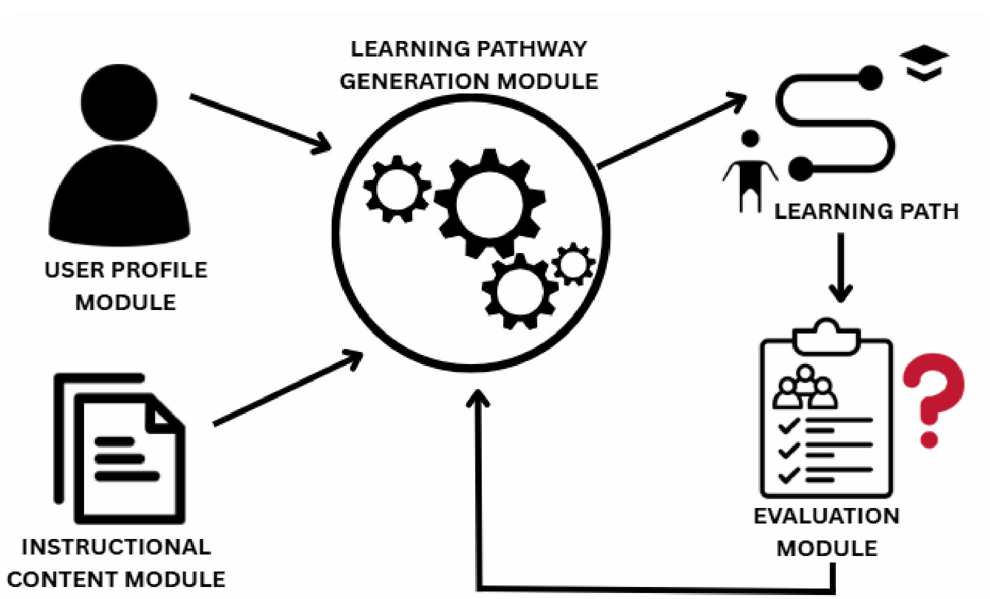

Figure 4.1: Overview of the architecture

4.1 USER PROFILE MODULE

User profiling aims to build a comprehensive cognitive profile for each student. As discussed in paper [12], relying solely on self-reported profiles is inadequate for accurately assessing cognitive abilities. Therefore, it is essential to incorporate assessment methods that evaluate specific metrics. Based on our review of the literature, the following metrics have been identified as critical for user profiling:

- Reading capabilities
- Working memory capacity
- Information processing abilities
- Learning styles according to the VARK model

The user profiling strategies have been adapted from the papers [7] and [11]. These metrics form the foundation for personalizing the learning path.

4.1.1 Reading Capabilities

Paper [11] emphasizes the effectiveness of the Flesch Reading Ease (FRE) score in adapting content to match a student's individual writing style, thereby minimizing potential interference during the learning experience. Consequently, the student is provided content that better matches their reading capability. A student's reading capability is assessed using the FRE metric, which is computed as:

$$\text{Reading Ease Score} = 206.835 - 1.015 \times \left(\frac{\text{tw}}{\text{ts}}\right) - 84.6 \times \left(\frac{\text{tsl}}{\text{tw}}\right) \tag{4.1}$$

where:

- tw denotes the total number of words,
- ts denotes the total number of sentences,
- tsl denotes the total number of syllables.

Students are asked to compose a short passage on a topic of their choice. The FRE score of the submitted text is then calculated to evaluate their reading abilities.

Levels of Flesch-Kincaid Reading Ease Score (0–100):

- **90–100**: Very easy to read, suitable for younger readers or simple instructions.
- **60–80**: Plain English, ideal for most web content.
- **30–50**: Fairly difficult to read.
- **0–30**: Very difficult to read.

The resulting score typically ranges from 0 to 100, where a higher score indicates easier readability. The score is rounded to the nearest integer to obtain the readability score, r_d.

4.1.2 Working Memory Capacity

Working memory capacity is evaluated using an N-back test, as introduced in paper [11]. Papers [13] and [14] highlight the effectiveness of the assessment. The assessment consists of a three-part N-back test involving visual, auditory, and textual stimuli. For each stimulus type, the student must identify whether the current item matches the one presented n trials earlier. Scores for each type of stimulus are recorded on a scale of [0,5], resulting in wm_v, wm_a, and wm_t for visual, auditory, and textual stimuli, respectively. We specifically employ the 2-back test, as 1-back tasks tend to produce ceiling effects, while 3-back tasks often result in flooring effects, as noted in paper [13]. Therefore, the 2-back test is deemed the most suitable choice for this assessment. It is important to note that there are no standardized scoring benchmarks for the N-back task; the scores are relative and primarily serve to differentiate individuals based on their working memory capacity within the given experimental context.

4.1.3 Information Processing

Information processing ability is assessed using Bloom's taxonomy, a cognitive framework comprising six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create, as introduced in paper [8]. The assessment is divided into three sections based on the mode of information: visual, auditory, and textual. Scores for each mode are recorded on a scale of [0,6], represented by ip_v, ip_a, and ip_t for visual, auditory, and textual stimuli, respectively. Fig. 4.2 illustrates the different levels of Bloom's taxonomy.

4.1.4 Learning Style

The learning styles attributed to the students are based on the VARK model introduced in papers [7, 11]. However, this model has been adapted for our e-learning platform by

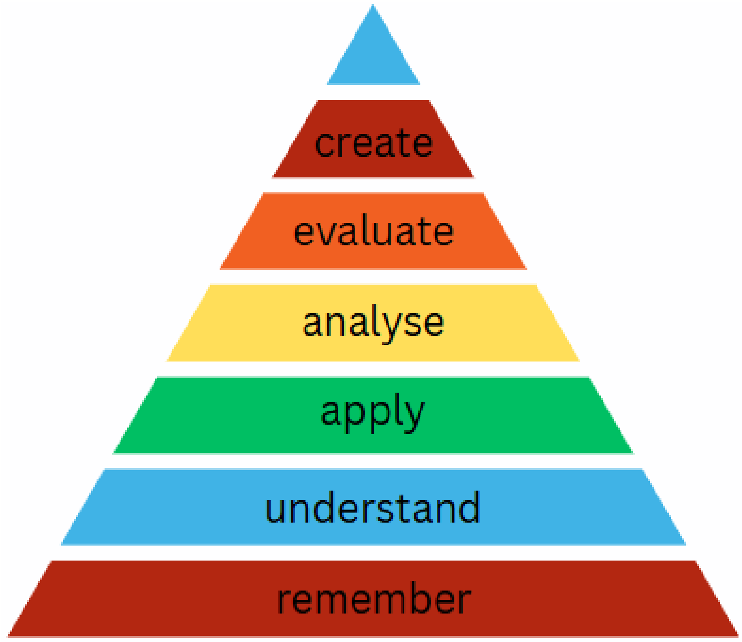

Figure 4.2: Bloom's taxonomy

eliminating the kinesthetic component, which is less applicable in an online environment. Instead of relying on self-reported questionnaires, learning styles are indirectly inferred from the working memory capacity and information processing assessments. These assessments were further divided into three categories—visual, auditory, and textual—to derive corresponding scores for each modality. The student's learning style is determined by combining the corresponding scores, resulting in visual (v), auditory (a), and textual (t) learning style scores computed as follows:

$$s = wm_v + wm_a + wm_t + ip_v + ip_a + ip_t \tag{4.2}$$

$$v = \frac{wm_v + ip_v}{s}, \quad a = \frac{wm_a + ip_a}{s}, \quad t = \frac{wm_t + ip_t}{s} \tag{4.3}$$

The resulting values are normalized such that $v + a + t = 1$.

4.2 INSTRUCTIONAL CONTENT MODULE

The Instructional Content module is responsible for managing the learning content to be delivered to students, along with its associated metadata. Content encompasses various formats such as video, audio, text, games, or gamified material, chosen based on the learner's preferences and needs.

As introduced in the domain model of paper [4], instructional content is organized into

Learning Objects (LOs), each annotated with the following metadata:

- **Readability (r)**: Measured using the Flesch Reading Ease score for text, within the range [0,100].
- **Learning Style Scores (v_c, a_c, t_c)**: Ratings (0–5) indicating the compatibility of the content with visual, auditory, and textual learning styles, ensuring that $v_c + a_c + t_c = 5$.
- **Information Processing (ip)**: A Bloom's taxonomy score assigned to each topic, ranging from [0,6].
- **Average Difficulty (d)**: A difficulty rating assigned to each topic, ranging from [0,5].

References [7], [8] and [11] provide strategies for extracting metadata from the LOs. The metadata extraction is elaborated in the Implementation (chapter 5). The learning style scores, information processing score, and readability score collectively enable the mapping of suitable Learning Objects (LOs) to individual students.

4.3 LEARNING PATHWAY GENERATION MODULE

This module is responsible for generating a personalized learning sequence for the course. It utilizes a course graph, represented as a directed acyclic graph (DAG), which captures the dependencies and relationships between topics to produce a coherent sequence. Genetic Algorithms (GA) is a search and optimization technique inspired by the process of natural selection in biology. It mimics the principles of evolution—selection, crossover, and mutation—to iteratively evolve better solutions to a given problem. GA is employed here to map appropriate Learning Objects (LOs) to each student profile, with the course graph as a constraint to the learning path. The components of the GA are described below. Fig. 4.3 illustrates the overall workflow of the module.

4.3.1 Gene

A gene represents the smallest unit of information within a chromosome. Each LO with associated metadata is represented as a gene. Fig. 4.4 shows a gene.

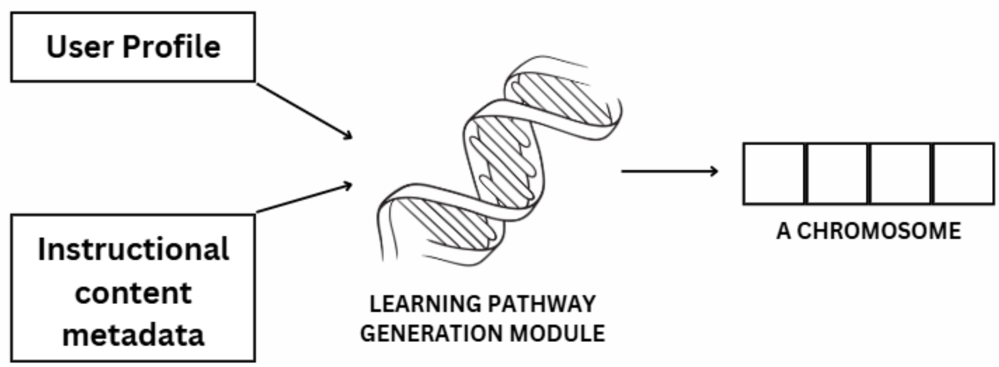

Figure 4.3: Learning pathway generation module

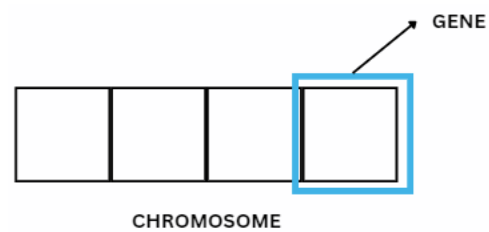

Figure 4.4: Gene and chromosome in a course sequence

4.3.2 Chromosome

A chromosome is an ordered sequence of genes that together define a complete learning path for a student. The final chromosome after GA execution represents an optimized learning sequence. Fig. 4.4 shows a chromosome and a gene within a chromosome.

4.3.3 Selection

Selection is the process of choosing candidate chromosomes from the current population for the next generation. Common selection methods include tournament, roulette wheel, and rank selection. The selection strategy is experimentally determined on the basis of the fitness values obtained with each generation. The experimental comparison and justification for the chosen selection strategy are detailed in Section 6.2.1.

4.3.4 Crossover

Crossover involves combining parts of two parent chromosomes to produce one or more offspring. This operation mimics biological reproduction and allows the exchange of genetic material, enabling the algorithm to explore new parts of the solution space.

In this thesis, a partially mapped crossover (PMX) is employed, as introduced in paper [8]. Paper [17] highlights the effectiveness of PMX in maintaining logical ordering and avoiding repetition during crossover operations.

In PMX, a random crossover section is selected, and the genetic material within this section is exchanged between two parent chromosomes. A mapping relationship is established between corresponding elements, ensuring that offspring are valid and maintain topic dependencies. Fig. 4.5 illustrates the PMX crossover applied to two parental chromosomes, where A, B, etc., represent course topics and different colors denote specific LOs associated with each topic.

To better suit the course structure, a modification of PMX is implemented, producing **only one offspring** with each crossover operation. After the crossover section is exchanged, the algorithm checks for duplicate topics in the offspring. Genes are filled outside the crossover segment by selecting learning objects that do not repeat topics already present within the child chromosome. This ensures that each topic appears only once, leading to coherent and diverse offspring.

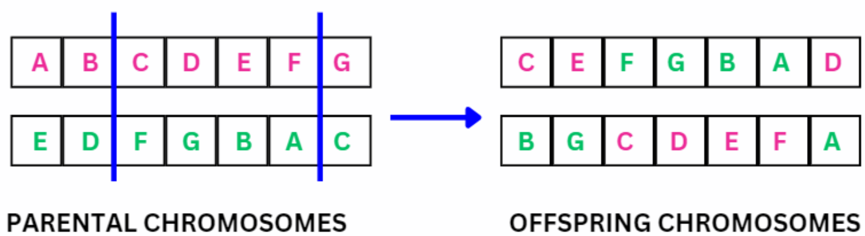

Figure 4.5: Partially mapped crossover (PMX)

4.3.5 Mutation

Mutation introduces diversity into the population by replacing a gene (i.e., a learning object, LO) with another LO corresponding to the same topic. The mutation rate is experimentally determined to balance exploration and convergence. The selection of the mutation rate and its impact on performance are discussed in detail in Section 6.2.2.

Fig. 4.6 illustrates the mutation process within a chromosome, where the numbers 1, 2, 3, and 4 represent topics in the course, and each color denotes a specific LO associated with the respective topic.

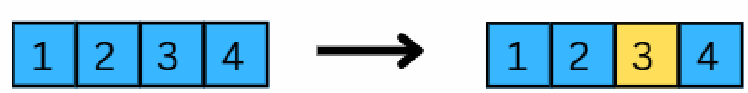

Figure 4.6: Mutation

4.3.6 Gene Repair

To ensure that the generated learning path follows a meaningful sequence of topics, a dependency graph is provided as input. This graph is modeled as a directed acyclic graph (DAG) that captures the relationships between the various topics in the course. To maintain consistency between the generated path and the dependency structure, a gene repair operator, introduced in paper [18], is incorporated. The paper highlights how the operator results in faster GA convergence. Specifically, we employ a Layered Topological Sort, as described in paper [8], to enforce a topological order that respects the prerequisite relationships defined in the course graph.

The definition of a Layered topological sort is as follows: Let R be a partially ordered relation on the vertex set

$$M = \{a_1, a_2, \ldots, a_n\},$$

where a_i represents a vertex in M. For each vertex a_i, there exists a weight l_i, indicating its precedence in the topological order. The vertex sequence A is defined as:

$$A = \{(a_1, l_1), (a_2, l_2), \ldots, (a_n, l_n)\}.$$

Each pair (a_i, l_i) is referred to as a *binary combination*. For $0 \leq i \leq n-1$, for each two adjacent binary combinations (a_i, l_i) and (a_{i+1}, l_{i+1}), the sequence A is considered to be in *layer-topological-order* relative to the relation set R if:

$$0 \leq l_{i+1} - l_i \leq 1.$$

The weight l_i is defined as the layer in the DAG. Vertices with the same weight are exchangeable in the vertex sequence A.

Furthermore, within a given layer of the DAG, topics with **lower difficulty** values are prioritized in the sequencing. This strategy ensures that if a student encounters difficulties with a particular topic, the algorithm can restructure the learning path by temporarily moving the student to another accessible topic, and subsequently revisiting the challenging topic at a later stage.

Consider the course graph given in Fig. 4.7. An accepted sequence of topics in a chromosome for this graph is: A, D, B, E, C, F, G

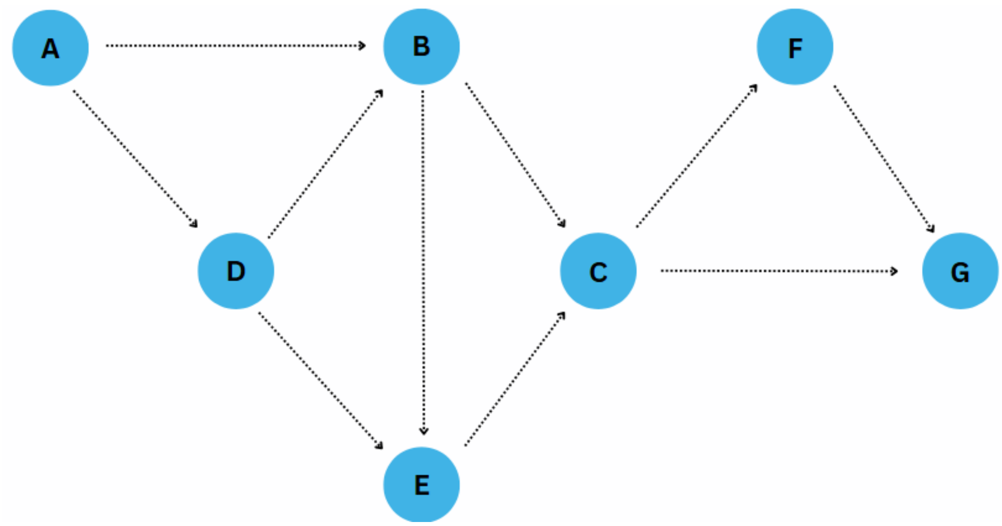

Figure 4.7: Course graph

4.3.7 Fitness Function

The fitness function is formulated based on the concept of **learning cost** associated with the learning objects (LOs), and it serves as the foundation for the genetic algorithm (GA). It has been adapted from paper [8], adjusting the formula to align with our defined metrics. A fitness value is assigned to each chromosome within a generation to evaluate its suitability for the student profile. The chromosome with the best fitness value explored during the evolution process is selected, and the corresponding sequence of learning objects is delivered to the student.

The fitness function can be interpreted as the total learning cost of a chromosome, where a lower cost indicates better fitness. For a given gene, the learning cost l_g is formulated as:

$$l_g = d + w_r + t \cdot (5 - t_c) + a \cdot (5 - a_c) + v \cdot (5 - v_c) + w_{ip} \cdot ip \tag{4.4}$$

where,

- l_g: learning cost of a gene [0-33]
- d: difficulty of the LO represented by the gene $[0 - 5]$
- w_r: weight for readability [0-5]
- t: preference of textual learning style of the student $[0 - 1]$
- a: preference of auditory learning style of the student $[0 - 1]$
- v: preference of visual learning style of the student $[0 - 1]$

- t_c: amount of textual content in the gene $[0-5]$
- a_c: amount of auditory content in the gene $[0-5]$
- v_c: amount of visual content in the gene $[0-5]$
- w_{ip}: weight for information processing [0-1]
- ip: level of information processing required for the gene [0-6]

The values d, t_c, a_c, v_c, and ip are obtained from the LO metadata. t, a, and v are obtained from the student profile. The weights w_r and w_{ip} are calculated as given below:

$$w_r = \begin{cases} 0, & \text{if } rd < r, \\ \frac{rd-r}{20}, & \text{otherwise.} \end{cases} \tag{4.5}$$

where:

- r: readability ease of the content,
- rd: readability score of the student.

The values rd and r are obtained from the student profile and LO metadata, respectively. If the LO is more readable than the reading capability of the student, then the cost is 0. Otherwise, the difference is the learning cost. Note that the maximum combined score from the working memory and ip test is 33.

$$w_{ip} = 1 - \frac{s}{33} \tag{4.6}$$

Here, w_{ip} represents the combined score of information processing and working memory capacity of the student and s is defined in Equation. 4.2.

For a chromosome $C = (g_1, g_2, \ldots, g_n)$, the learning cost lc is formulated as:

$$lc = \frac{1}{n} \sum_{i=1}^{n} l_{g_i} \tag{4.7}$$

The local learning cost l'_i of a pair of genes is given by:

$$l'_i = \frac{l_{g_{i-1}} + l_{g_i}}{2}, \quad i = 2, 3, \ldots, n \tag{4.8}$$

The fitness value f of a chromosome is given by the learning cost of the chromosome and the mean square deviation of the local learning costs of the genes.

$$f = lc + \sum_{i=2}^{n} (l'_i - lc)^2 \tag{4.9}$$

To explore the impact of simpler formulations, we also experimented by using just the mean learning cost lc as the fitness value instead of the full fitness formulation. The mean was chosen due to its computational simplicity and interpretability, with further justification provided in Section 6.3. This approach yielded reasonable results and served as a useful baseline for comparison.

4.3.8 Generation process

The following steps outline how a learning path (chromosome) of a course is generated for a given student profile using the GA functions discussed so far. Fig. 4.8 graphically depicts the steps in the process.

1. **Initial Population:** A population of random chromosomes is generated. Each chromosome consists of exactly one gene for each topic in the course. For a course with n topics, each chromosome contains n genes. This serves as the 0th generation of the algorithm.

2. **Selection:** A subset of the population is selected based on fitness values. Fitter individuals have a higher probability of selection.

3. **Crossover:** Selected individuals undergo crossover to produce offspring. Genetic material is exchanged between two parent chromosomes.

4. **Mutation:** Mutation is applied to each offspring with a predefined probability. A random gene in the chromosome is replaced with another gene of the same topic.

5. **Gene Repair:** It ensures that each chromosome maintains a valid and meaningful sequence, fixing any inconsistencies introduced during crossover and mutation.

6. **Next Generation:** The newly obtained chromosomes form the next generation. The algorithm runs for a specified number of generations to identify the most suitable chromosome for the student.

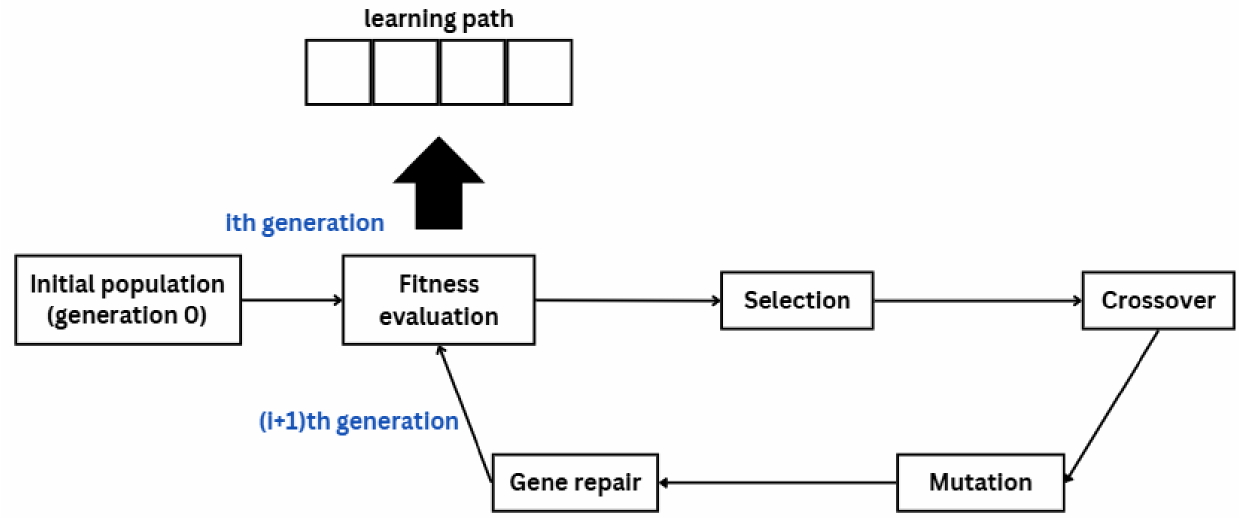

Figure 4.8: Learning path generation process

4.4 OPTIMIZATION STRATEGIES

For every student, the algorithm begins with a group of random chromosomes as the 0th generation. Creating a better starting point for the algorithm can provide a head start and improve the efficiency of the search process. References [15, 16] highlight the advantages of clustering in e-learning and personalization. Clustering is, thus, employed to group students on the basis of some learning characteristics, which would additionally provide a better initial population. To align with our objective, clustering is employed to group students with similar learning styles, enabling the formation of a more relevant gene pool for the GA. It facilitates grouping students with similar learning profiles, enabling the creation of specialized initial populations for the GA. This not only accelerates convergence but also enhances the relevance and effectiveness of the generated learning paths by aligning them more closely with individual learner needs.

- **Cluster Representation:**
  - A cluster is denoted by a tuple $\{a, b, c\}$, representing its centroid.
  - Each cluster maintains a gene pool.
- **Student Representation:**
  - A student is represented by their profile, a tuple $\{v, a, t\}$, which corresponds to their Visual, Auditory, and Textual learning scores based on the VARK model.
- **Cluster Assignment:**

- The distance between the student profile and each cluster centroid is calculated. Fig. 4.9 depicts this scenario.
- If the minimum distance is below a predefined threshold, the student is assigned to the closest cluster.
- The centroid of the assigned cluster is updated as the average of the previous centroid and the new student profile.

- **New Cluster Creation:**
  - If all cluster distances exceed the threshold, a new cluster is created.
  - The student profile serves as the centroid of the new cluster.

- **Generating the Initial Population:**
  - If the student is assigned to an existing cluster, the initial population is generated from that cluster's gene pool.
  - If a new cluster is created, the initial population is generated randomly.

- **Updating the Gene Pool:**
  - After the genetic algorithm runs and a chromosome is obtained, the assigned cluster's gene pool is updated.
  - The genes from the obtained chromosome are added to the cluster's gene pool.

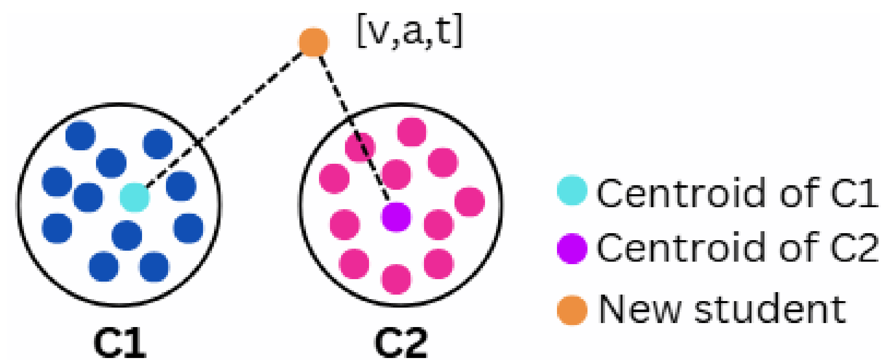

Figure 4.9: Clustering

4.5 EVALUATION MODULE

Once a personalized learning path is generated for the student, it is essential to evaluate the student's understanding at various stages of the course. To facilitate this, an assessment framework is implemented, as depicted in Fig. 4.10 , which outlines the overall structure of the evaluation process. The assessment process is divided into three key stages:

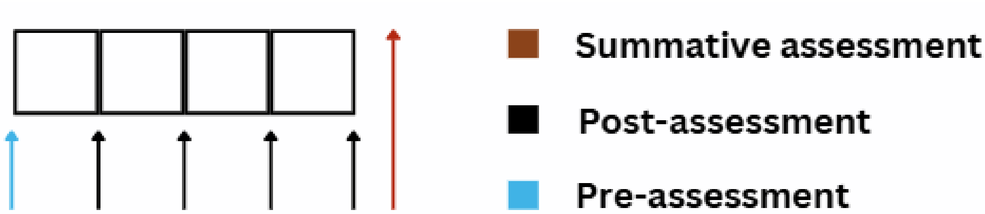

Figure 4.10: Assessment structure

1. **Pre-Assessment**: Before the student begins the course, a pre-assessment is conducted to gauge their prior knowledge and readiness for the course.

2. **Post-Assessment**: After each topic, a post-assessment evaluates the student's understanding of that topic. The following steps are taken based on the student's score:

   (a) If the student scores **80% or more**, they are considered to have passed the assessment. They then proceed to the next topic in the sequence. If the current topic is the final one, the student is deemed to have successfully completed the course. Fig. 4.11 illustrates this case.

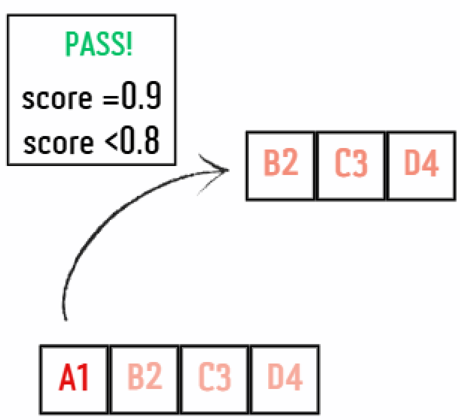

Figure 4.11: Post assessment passed

   (b) If the student scores **below 80%**, they are considered to have failed the assessment:

i. On the **first or second failed attempt**, the current learning resource (gene) is **replaced** with an alternative resource corresponding to the same topic, allowing the student to engage with different content for improved understanding. Fig. 4.12 illustrates this case.

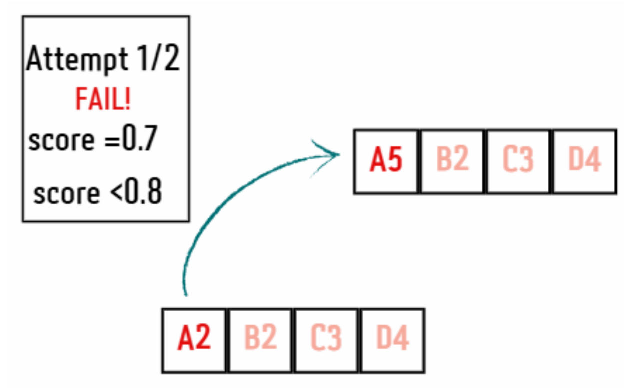

Figure 4.12: Post assessment failed - gene replaced

ii. For the **third failed attempt**, the **topic difficulty is increased** for the student. The chromosome is **regenerated** with modified difficulty values to allow easier topics to appear earlier in the sequence. The *gene repair operator* ensures correct sequencing.

   Two possible cases arise during this regeneration:

   A. If two topics (e.g., A and B) are on the same graph level and A is more difficult than B, then they can be swapped to allow the easier topic B to come first. This scenario is illustrated in Fig. 4.13.

   B. If A and B are on different levels (i.e., they have a dependency), they cannot be swapped, even if B has a lower difficulty. This ensures that the prerequisite structure of the course remains valid. This scenario is illustrated in Fig. 4.14.

iii. After regeneration, the student re-learns the modified topic and re-attempts the post-assessment.

(c) Based on the outcome of the reassessment, steps (a) or (b) are repeated until the student passes or completes the course.

3. **Summative Assessment**: At the end of the course, a summative assessment is performed to measure the student's overall understanding of the course content. This provides a general measure of learning outcomes and validates the effectiveness of the personalized learning path.

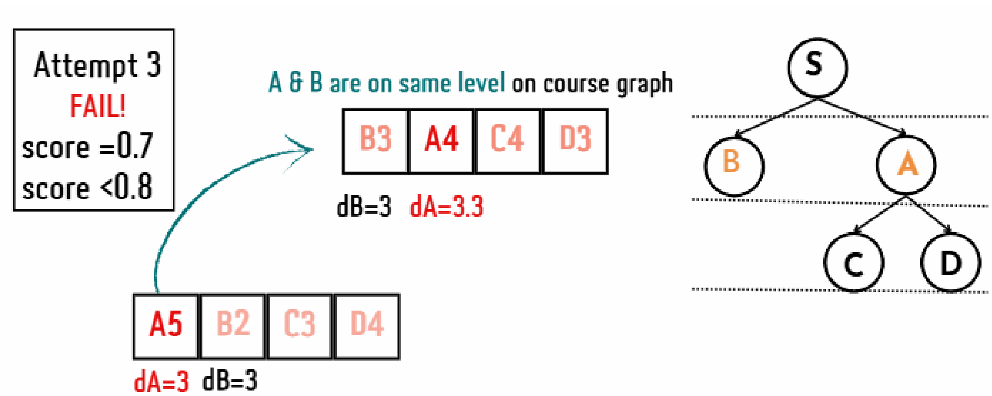

Figure 4.13: Path regenerated with increased difficulty and change of sequence

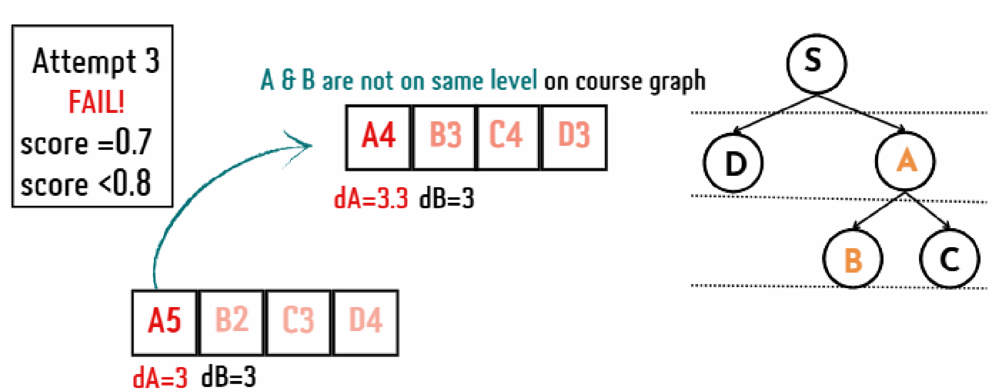

Figure 4.14: Post assessment failed - path regenerated with increased difficulty

4.6 CONCLUSION

The methodology adopted in this project brings together cognitive modeling, content structuring, and evolutionary computation to build an adaptive and personalized e-learning framework. **The User Profile Module** lays the foundation by assessing students' reading capabilities, working memory capacity, and information processing skills. These metrics are further used to infer the learning style of each student, categorized into visual, auditory, and textual modalities. Building on this profile, the **Learning Pathway Generation Module** constructs personalized learning sequences using the principles of Genetic Algorithms. The evolutionary process involves selection of fit pathways, crossover to combine promising solutions, and mutation to introduce diversity. A gene repair mechanism ensures that generated pathways remain valid, and a fitness function evaluates each pathway based on how well it aligns with the learner's profile. This cycle repeats across generations to optimize learning pathways. The **Instructional Content Module** supports this system by organizing and tagging learning materials with appropriate metadata and implementing optimization strategies to ensure content relevance and sequencing. Finally, the **Evaluation Module** monitors learner engagement and progress, enabling ongoing refinement of the learning experience. Together, these components create a dynamic and intelligent learning environment tailored to individual cognitive strengths and preferences, with the flexibility to evolve and scale for broader use.

The following chapter delves into the implementation details of the proposed methodology. It provides a comprehensive overview of how each module has been realized, the technologies and tools used, and how the various components interact in a real-world system. It also elaborates on the experimental setup and the practical considerations taken into account while building the adaptive learning platform.

Chapter 5

Implementation

This chapter details the implementation of the proposed framework. To assess the feasibility of the methodology, a proof of concept was initially developed. Building on this, a prototype was created to demonstrate the core functionalities of the framework. The following sections provide a detailed explanation of these components.

5.1 PROOF OF CONCEPT

A proof of concept has been developed using **Google Colab** to demonstrate the intended functionality of the system. Given a student profile and a repository of instructional content annotated with metadata, the implementation illustrates the algorithm workflow and generates the expected output. Fig. 5.1 presents the course graph employed in the proof of concept, including the arbitrarily assigned difficulty levels for each topic in the course.

Note: In this implementation, each Learning Object (LO) has been assigned an arbitrary information processing (ip) score. This assignment will be revised in the following sections, where the ip score will instead be associated with the respective **topic** rather than individual LOs.

You can find the GitHub link to this Colab notebook here: `https://github.com/Sreeshu123/Final-year-project--GA`

5.1.1 Learning Path Generation Using Genetic Algorithms

The **PyGAD** library of Python has been chosen to perform the GA functions, owing to its flexibility in customizing core genetic operations and ease of integration into prototype systems. It provides essential GA features while allowing the definition of tailored crossover, mutation, and fitness strategies, making it suitable for research and experimentation in

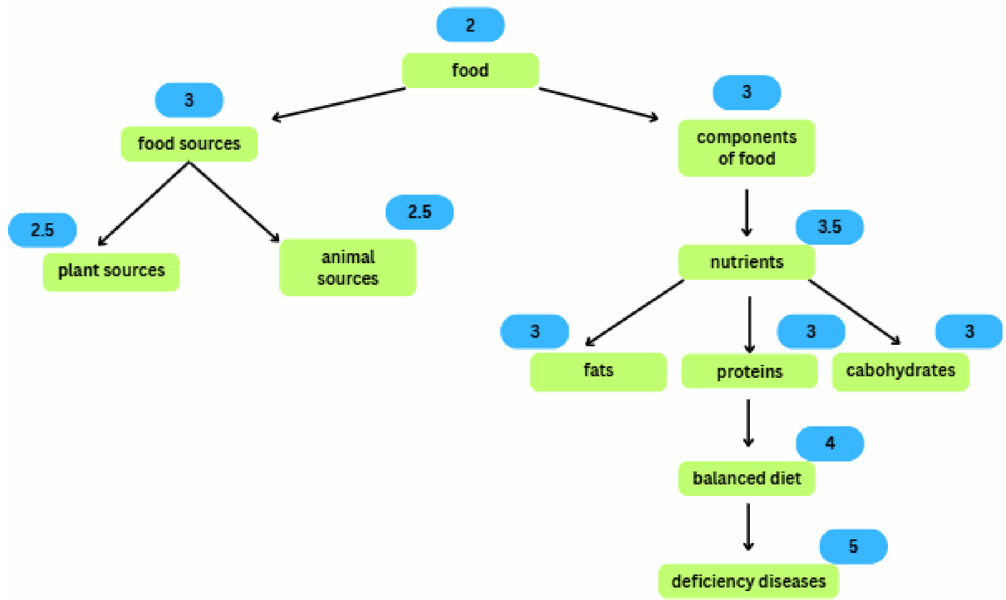

Figure 5.1: Course graph of proof of concept

adaptive learning contexts. Each of the Genetic Algorithm (GA) functions is implemented as follows:

- **Selection:** Built-in selection methods of PyGAD were used for experimentation, including tournament, rank, and roulette wheel.
- **Crossover:** A custom PMX crossover function was designed as per the theory given under `crossover` of Section 4.3.
- **Mutation:** A custom mutation function was designed as per the theory given under `mutation` of Section 4.3.
- **Gene Repair:** A custom gene repair function was designed as per the theory given under the gene repair section of Section 4.3. As PyGAD does not include gene repair functionality by default, it is included under the `on_generation` callback function.
- **Fitness Evaluation:** A custom fitness function was designed as per the theory given under the fitness function section of the methodology. PyGAD by default maximizes the fitness value. However, here the fitness value represents the learning cost, which must be minimized. Hence, the fitness values are **negated**.
- **Keep Elitism:** This is an in-built feature of PyGAD that carries the best solutions in the current generation to the next generation. This is set to 1, meaning the best solution in the current generation is propagated.

Each generation of the algorithm consists of **16 individual chromosomes**.

A GA instance is initialized with the above custom functions to generate the learning path. The instance takes an initial population as input and outputs a path of instructional content suitable for the student profile. Fig. 5.2 shows the evolution of fitness values during the course of the genetic algorithms.

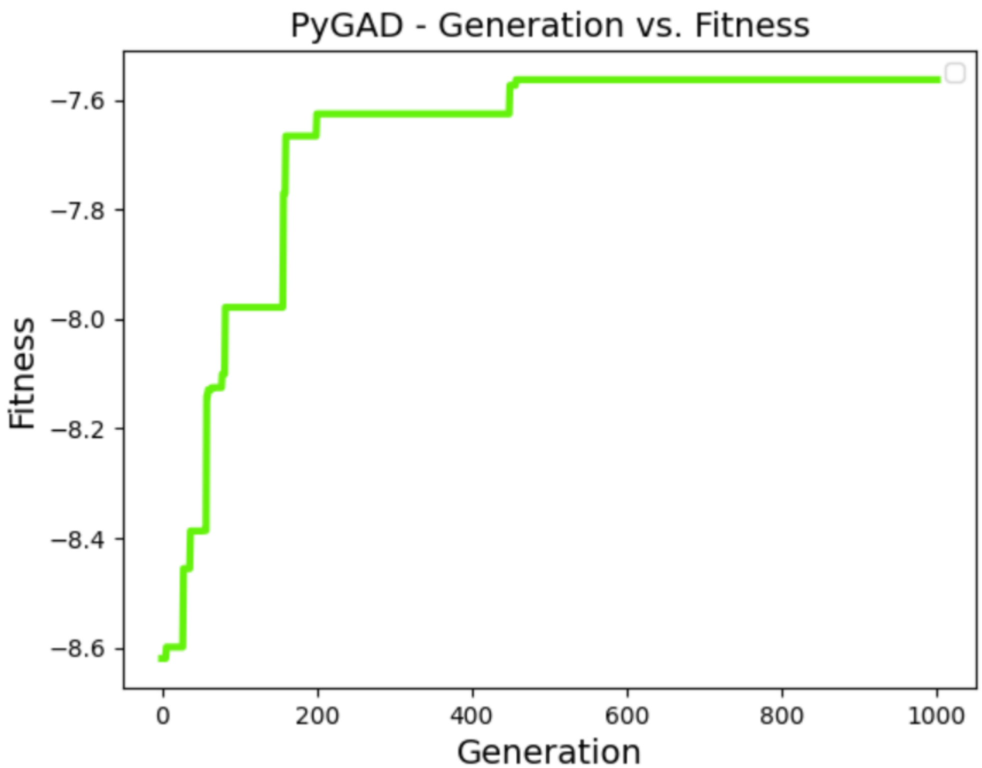

Figure 5.2: Fitness value evolution over the generations

5.1.2 Clustering

In order to demonstrate the creation of clusters, dedicated data structures were initialized to store them. The following functions were implemented to handle clustering-related operations:

- **cluster_assignment:** Assigns a student to a cluster based on the strategy described in Setion 4.4.
- **add_path:** Updates the gene pool of the assigned cluster, following the logic outlined in Section 4.4.

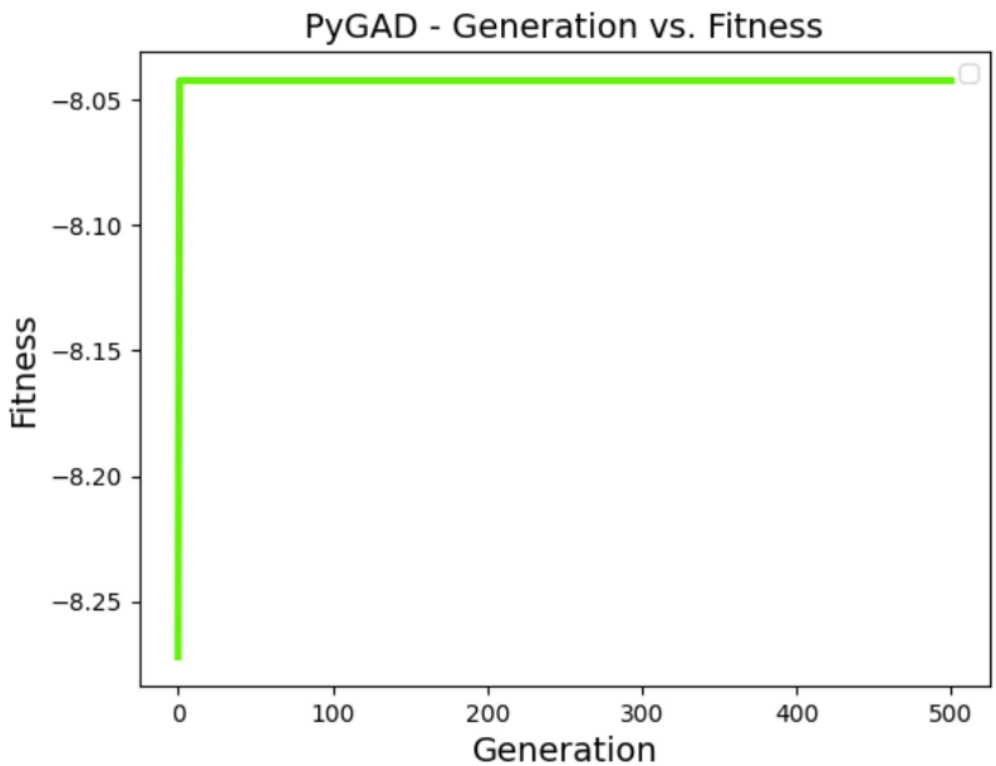

Figure 5.3: Fitness value evolution using clustering

Fig. 5.3 illustrates the fitness evolution with clustering enabled. It highlights how initializing the population from the cluster's gene pool provides a more suitable starting point for the student, thereby reducing the number of generations required to converge on an optimal learning path.

5.1.3 Post-assessment

The proof of concept illustrates the structure of post-assessments, showcasing how the learning path dynamically adapts in response to the quiz score input. A simulated quiz score is generated using Python's `random` library. The `update_score` function then modifies the learning path accordingly. This reflects the system's dynamic response to learner performance, as outlined in Section 4.5, which details the underlying adaptation logic.

5.1.4 Conclusion

A proof of concept was developed using **Google Colab** to validate the feasibility of the proposed methodology. The **PyGAD** library in Python was utilized to implement the genetic algorithm components. A randomly generated user profile and a set of learning objects (LOs) with associated metadata were used to simulate the system. The implementation demonstrates the complete learning path generation process, both with and without clustering. Post-assessment path updates were simulated using Python's `random` library.

This effort has resulted in a functional code base that serves as a strong foundation for developing a fully operational personalized e-learning system. To validate the proposed methodology, a set of instructional content was curated and annotated with metadata indicating whether it was primarily visual, auditory, or textual. Correspondingly, three student profiles—each representing a dominant learning style (visual, auditory, and textual)—were stored in the system. The algorithm was then executed, and it successfully mapped the appropriate content to each student based on their individual learning preferences, thereby demonstrating the intended behavior of the personalization logic. This experiment highlights the effectiveness of the proposed framework in aligning content delivery with user-specific cognitive traits, establishing its feasibility for real-world application.

The proof of concept, however, revealed that extending the implementation to fully support neurodiverse students presents several challenges. It requires extensive modifications to the user interface to ensure accessibility, usability, and comfort for a wide range of cognitive and sensory preferences. Moreover, neurodiversity encompasses a broad spectrum of differences, making it difficult to accommodate every need within a single implementation. As a result, the current system focuses primarily on personalization based on cognitive profiling, while comprehensive support for neurodiverse learners is identified as an important direction for future work.

The focus of the upcoming prototype section is on the implementation of the following components, which are to be integrated with the existing code base developed during the proof of concept:

- **User Profiling through Assessments:** Designing and implementing initial assessments to evaluate students' cognitive traits, including working memory, information processing, and preferred learning style.

- **Instructional Content Acquisition with Metadata:** Curating and organizing instructional content, annotated with relevant metadata such as modality (visual, auditory, textual), readability, difficulty level, and cognitive demands.

- **Evaluation Module:** Implementing quizzes and evaluations after each learning object to assess student understanding, update performance metrics, and iteratively refine the personalized learning path.

In addition to these components, the prototype also involves integrating the existing code base with a Learning Management System (LMS), to facilitate real-time content delivery and interaction within a structured course environment.

5.2 PROTOTYPE

The prototype of the project is implemented on **Moodle**, serving as an extension of the proof of concept developed earlier. **Moodle** was chosen as it is a widely adopted open source Learning Management System (LMS) that supports modular development, user management, and integration with external tools, making it ideal for testing personalized e-learning frameworks in realistic academic settings.

The moodle system is hosted on a cloud server and is available at `http://144.24.155.112/moodle/login/index.php`. Detailed documentation regarding the prototype is available at `https://docs.meetthecreator.in/`. A course has been created for the project, allowing users to enroll in the course. For each user, the following features are visible on the course page:

- Profiling assessment
- Pre-assessment
- Course pathway
- Summative assessment
- Learning style dashboard
- Learning progress dashboard

Fig. 5.4 highlights the features available on the course page.

Before proceeding to the section-wise implementation details, this section elaborates on the database schema and file structure of the Moodle project.

5.2.1 Database

This section details the database tables used in the project. There are two databases:

- **moodle**: Default database created by moodle to store moodle data
- **fyp**: Database created to store required information of the implementation

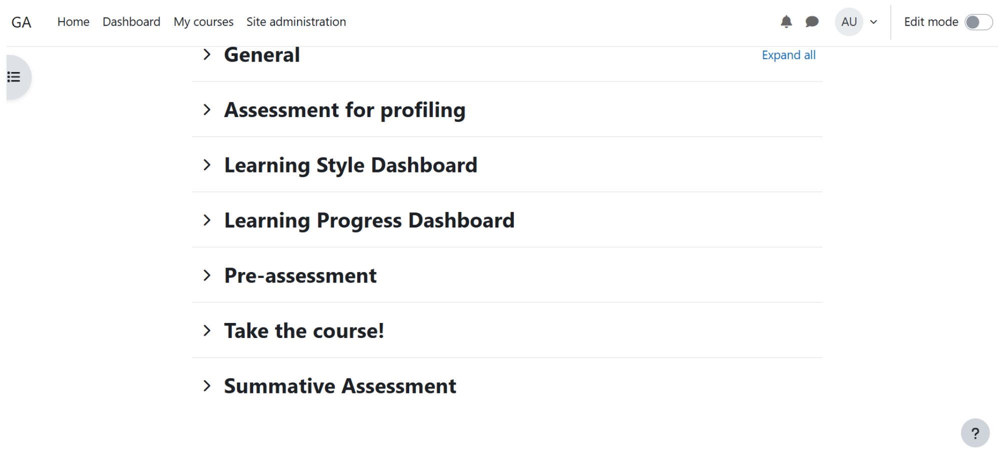

Figure 5.4: Course Page

Database: moodle

Note: moodle tables have many built-in fields. The schema of only the required section of tables is provided.

Table 5.1: mdl_user

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
flesch	INT	Flesch-kincaid score
ipv	FLOAT	Information processing score for visual content
ipa	FLOAT	Information processing score for auditory content
ipt	FLOAT	Information processing score for textual content
wmv	FLOAT	Working memory score for visual content
wma	FLOAT	Working memory score for auditory content
wmt	FLOAT	Working memory score for textual content
attempt	SMALL INT	Indicates whether mdl_poll_user.py has generated a path for the user

Table 5.2: mdl_questions

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
questiontext	TEXT	Text of the question
generalfeedback	TEXT	Feedback to be given with solution
qtype	TEXT	Indicates question type - multi-choice/truefalse

Table 5.3: mdl_question_answers

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
question	INT	Foreign key - id in mdl_questions
answer	TEXT	Choices for the question
fraction	FLOAT	Score of the choice

Table 5.4: mdl_question_versions

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
questionbankentryid	INT	Foreign key - id in mdl_question_bank_entries
questionid	INT	Foreign key - id in mdl_questions

Table 5.5: mdl_question_bank_entries

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
questioncategoryid	INT	Category of the question in the question bank

Database: fyp

Table 5.6: topics

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
topic_name	VARCHAR	Name of the topic
difficulty	FLOAT	Difficulty value of the topic
num_questions	INT	Number of quiz questions to be given for the topic

Table 5.7: lo

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
topic	INT	Foreign key - id in topics
v	FLOAT	Visual score
a	FLOAT	Auditory score
t	FLOAT	Textual score
ip	FLOAT	Information processing score
read_metric	FLOAT	Readability score
path	TEXT	Path to the resource in the server

Table 5.8: clusters

Field	Data type	Description
id	INT (AUTO INCREMENT)	Auto increment identifier
centroid	LONG TEXT	Centroid of the cluster as an array
gene_pool	LONG TEXT	Gene pool of the cluster as JSON

Table 5.9: students

Field	Data type	Description
user_id	INT	Foreign key - id in mdl_user
path	LONG TEXT	Current learning pathway of the student
cluster_id	INT	Foreign key - id of clusters
difficulty	LONG TEXT	Difficulty values topic-wise as JSON
gene_space	LONG TEXT	Gene space of the student as JSON
explored_genes	LONG TEXT	Genes explored by the GA in ascending order of learning costs and arranged topic-wise as JSON
learning_costs	LONG TEXT	Genes and their learning costs for the student as JSON
completed	LONG TEXT	Topics completed by the student as an array
stored_path	LONG TEXT	Initial learning path generated for the student

Table 5.10: quiz_attempts

Field	Data type	Description
student_id	INT	Foreign key - id in mdl_user
topic_id	INT	Foreign key - id in topics
attempts	INT	Quiz attempts of the user on the given topic
scores	LONG TEXT	Scores obtained by the student in the quiz attempts of the given topic

5.2.2 File Structure

The following files manage the complete process of generating the personalized learning path and delivering it to the student. These are encapsulated within the `learningpath` folder, which contains a custom Moodle plugin developed to provide the necessary functionalities.

Table 5.11: Files

File name	Path/folder	Input	Function
GA_functions.py	moodle/local/learningpath	None	Contains all GA-related and database access functions
script.py	moodle/local/learningpath	User id	Generates a personalized learning path for the given user and stores the values in table **students**
mdl_poll_user.py	moodle/local/learningpath	None	Triggers script.py when profile scores are updated in mdl_user
Quizupdate.py	moodle/local/learningpath	User id, topic id, score	Modifies the learning path after a quiz is completed and stores the updated path in **students**
scoring.py	moodle/local/learningpath	User id, topic id, score	Updates the quiz scores in table **quiz_attempts**
view.php	moodle/local/learningpath	Moodle user login	Displays course content to the student
get_next_lo.py	moodle/local/learningpath	User id	Fetches the next LO for the user from table **students**
quiz.php	moodle/local/learningpath	Moodle session data	Displays quiz for the previously taught topic
quiz_submit.php	moodle/local/learningpath	Moodle session data	Calculates quiz score and displays solutions
get_quiz_questions.py	moodle/local/learningpath	Topic id	Fetches topic-wise questions from **mdl_question**, **mdl_question_versions**, and **mdl_question_bank_entries**
dashboard.php	moodle/local/learningpath	Moodle user login	Displays graphical learning style score distribution
get_dashboard_data.py	moodle/local/learningpath	User id	Fetches user learning style scores for the dashboard
progress.php	moodle/local/learningpath	Moodle user login	Displays progress using quiz attempts and scores
get_progress_data.py	moodle/local/learningpath	User id	Fetches data about quiz attempts and scores

5.2.3 User Profile Module

User profiling aims to build a detailed cognitive and behavioral profile of each student based on a variety of learning-related metrics such as reading capabilities, working memory capacity, information processing abilities, and learning styles. These metrics serve as a foundation for tailoring the learning path.

This approach enables the system to better understand how individual students absorb, process, and retain information, thus creating the foundation for a personalized and adaptive learning experience. The goal is to tailor learning paths that align closely with each student's unique capabilities, preferences, and challenges.

A website was created to conduct the profiling assessments for students. The following sections explain how the scores were calculated. All these scores obtained are stored in the table **mdl_user**. Fig. 5.5, 5.6 and 5.7 show the website.

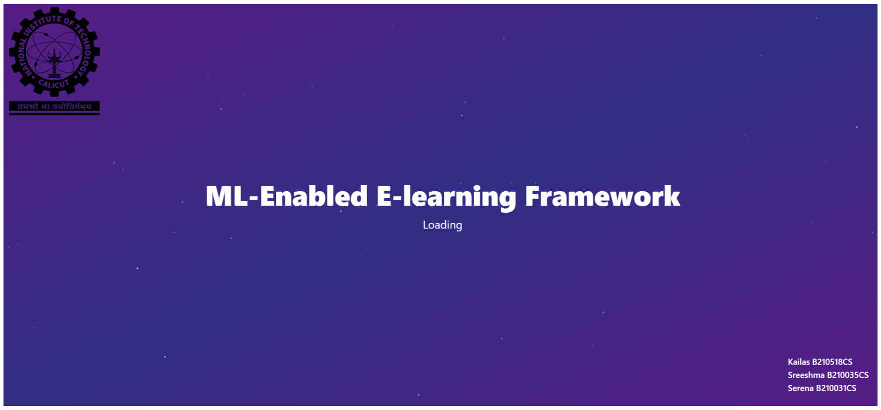

Figure 5.5: User profiling website landing page

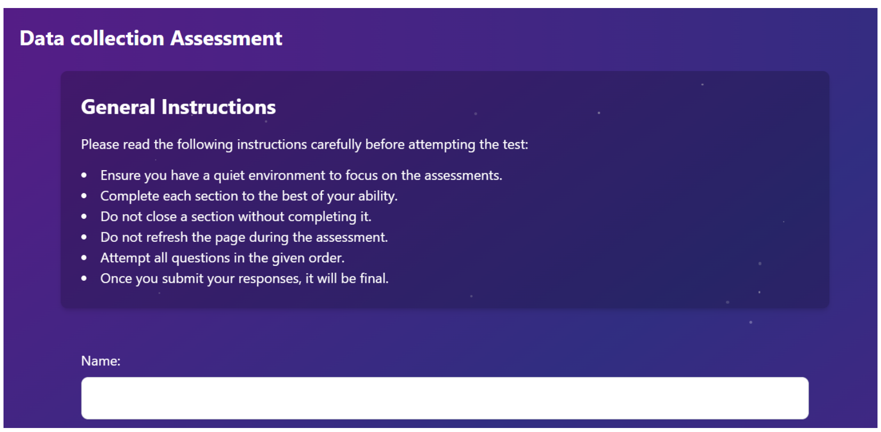

Figure 5.6: Assessment page

Reading Capabilities

The student's reading capability is assessed using the Flesch Reading Ease (FRE) metric. To obtain this score, the student is prompted to write a short paragraph on the topic 'Your School'. The FRE score is then computed based on equation 4.1. Fig. 5.8 shows the interface for the readability assessment section.

Working Memory Capacity

Based on the literature survey, 1-back test may lead to ceiling effects and 3-back test leads to flooring effects. Hence, working memory capacity is assessed using a three-part 2-back test

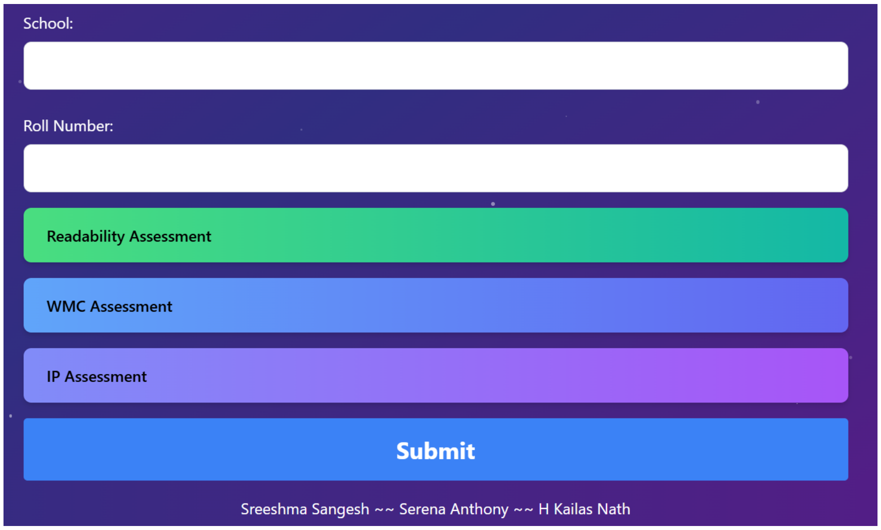

Figure 5.7: Assessment page

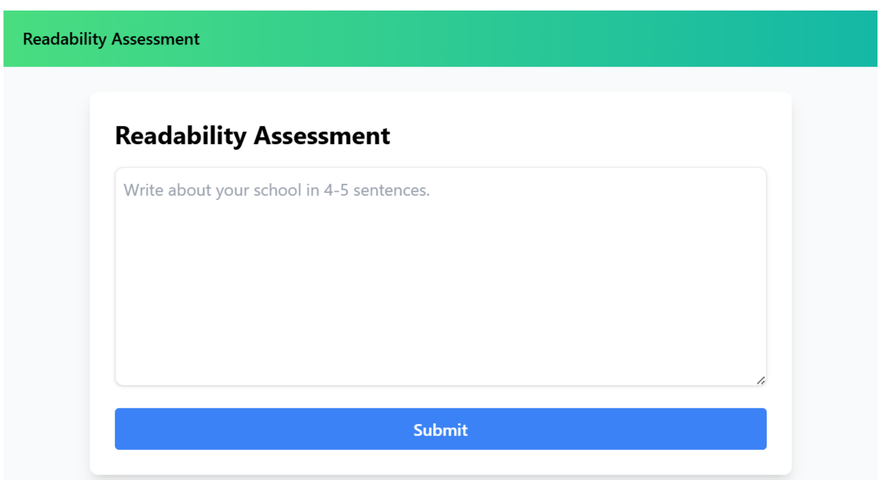

Figure 5.8: Readability Assessment

(visual, auditory, and text stimuli). For each type of stimulus, the student indicates if the current item matches the one presented 2-trials ago and scores were obtained as mentioned in section 4.1.2.

In the visual test, the students were shown a sequence of shapes such as triangle, circle, and square. Students were asked to identify whether the current shape matched the one displayed two images ago, thus testing their visual memory.

In the auditory test, the students were provided with a series of audio clips featuring vehicle names such as - bike, cycle, car and truck. Students were asked to identify if they heard the same name 2 clips ago, thus testing their auditory memory.

And finally in the textual test the students were presented with words representing fruits such as - apple, banana, cherry and kiwi. Students were asked to identify if the current word matched the one shown two steps earlier, thus testing their textual memory. Fig. 5.9 shows the interface for the working memory capacity assessment section.

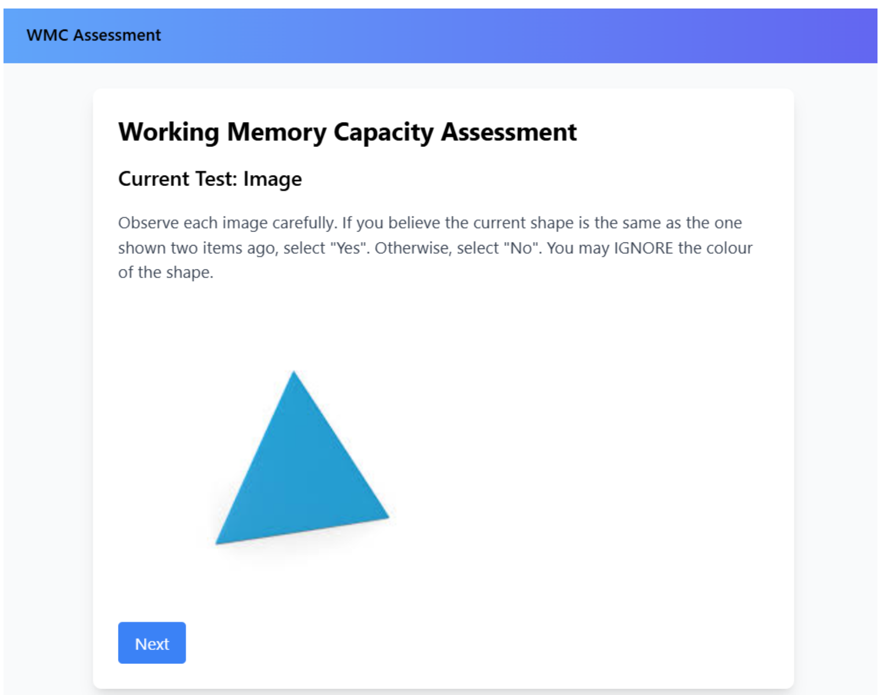

Figure 5.9: Working Memory Capacity Assessment

Information Processing

Information processing ability is evaluated using Bloom's taxonomy, which includes six cognitive levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. This assessment comprises of three parts - visual, auditory, and textual. For each part, there are questions from each level of the Bloom's taxonomy and scores were obtained as described in section 4.1.3.

The test is conducted to assess the students ability to process information across three sections: Textual, Visual, and Auditory. Each section presents a series of multiple choice questions (MCQ), carefully mapped to the six Bloom's levels.

In the textual assessment, students provided with a reading passage on the topic *'the water cycle'* and are required to answer MCQs based on the content, with the most appropriate answer. In the visual assessment, students are shown an image containing a sequence of shapes and answer MCQs based on it, with the most suitable answer. In the auditory test, students listen to an audio clip consisting of a sequence of sounds such as - thunder, a bell, car honking and birds chirping. They were then required to answer a set of MCQs based on the audio clip. Fig. 5.10 shows the interface of the information processing assessment section.

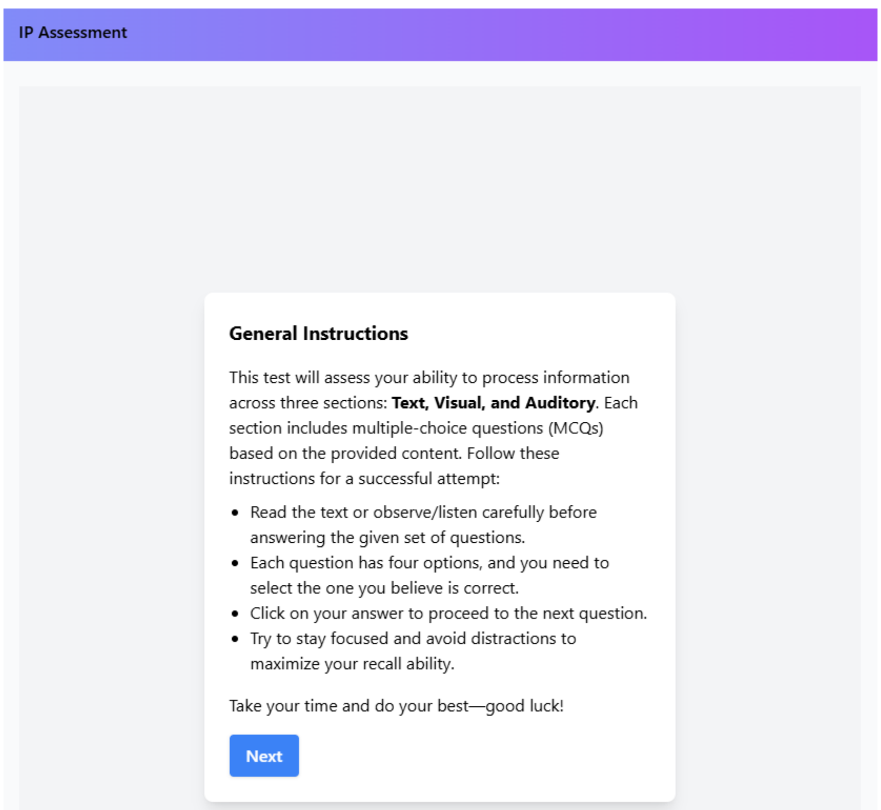

Figure 5.10: Information Processing Assessment

Learning Style

The Learning Style of each student is obtained from the profiling scores as outlined in section 4.1.4. These scores are calculated based on the student's performance across various modalities—visual, auditory, and textual. This personalized learning style profile plays a critical role in selecting and sequencing instructional content that best aligns with the student's cognitive strengths, thereby enhancing engagement and retention.

5.2.4 Instructional Content Module

The instructional content was based on the CBSE Class 6 NCERT Science textbook. Specifically, the chapters *"Food: Where Does It Come From?"* and *"Components of Food"* were selected. A total of **39 LOs** were collected, which primarily consisted of video and text-based resources. For each piece of instructional content, various metadata were calculated, including readability, learning style scores, information processing requirements, and average difficulty. The structure of the course content is as illustrated in Fig. 5.11. The videos and text materials used as instructional content were sourced from multiple websites, which have been credited in the Appendix.

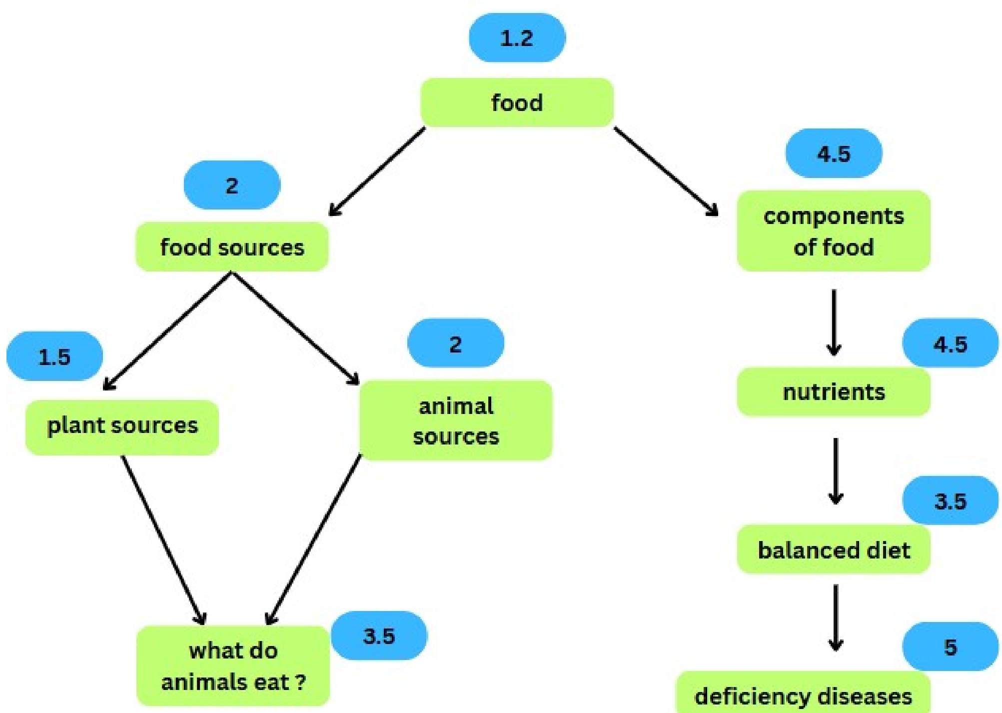

Figure 5.11: Course graph of chosen topics

Readability

To evaluate the readability of the textual materials, the **Flesch-Kincaid Readability Calculator** was employed, an established online tool designed to determine how easy a passage is to read. The tool considers factors such as average sentence length and the number of syllables per word to generate a readability score. For the **textual materials**,the readability of the content was obtained from the calculator. For the **video materials**, the text from the videos was extracted using online transcription tools. The transcriptions were manually verified to ensure correctness. The verified text was then fed to the readability score calculator. The obtained score is in the range [0,100]. The estimated reading grade levels based on the scores are summarized below.

Score	**Estimated Reading Grade Level**
90 to 100	5th grade
80 to 90	6th grade
70 to 80	7th grade
60 to 70	8th and 9th grade
50 to 60	10th to 12th grade (high school)
30 to 50	College
0 to 30	College graduate

Learning Style Scores

To evaluate the engagement and content quality of both video and text materials, a combination of Python libraries and multimedia processing techniques to compute detailed VAT (Visual-Audio-Textual) scores was employed.

For **video materials**, the VAT score was derived by individually analyzing visual, audio, and textual elements.

To assess the visual component of the videos, a combination of MoviePy, OpenCV, and a pretrained ResNet-18 model was employed. Frames were sampled at regular intervals across the video using MoviePy to ensure representative visual content is being considered. OpenCV, widely used for computer vision tasks, facilitated efficient frame extraction and enabled detailed analysis of motion intensity, shot transitions, brightness, and color richness—key indicators of visual engagement. In parallel, each extracted frame was passed through a pretrained ResNet-18 convolutional neural network, which produced a *confidence* *score* indicating the model's ability to classify the frame's visual content accurately. These confidence scores were averaged across all sampled frames to derive a quantitative measure of the video's visual interpretability. This combined approach provided insights into the clarity of the video content. And the final scores were obtained in the range [0,100].

To evaluate the auditory aspect of the video content, an audio intensity score was computed based on the Root Mean Square (RMS) energy, a widely accepted metric for measuring perceived loudness. The audio track was first extracted from the video using the MoviePy library. Subsequently, the audio waveform was processed using Librosa, a Python library for audio and music analysis. Librosa was used to calculate the RMS energy across short, evenly spaced segments of the audio. It reflects the overall energy and richness of the video's auditory component and serves as a quantitative indicator of how dynamically engaging the audio content is throughout the video. The final scores were obtained in the range [0,100].

To approximate the textual richness of video content, a lightweight proxy-based approach was used, leveraging the video's frame rate (frames per second, or FPS) as an indirect indicator of slide or text density. The frame rate was calculated by dividing the total number of frames by the video's duration. A higher FPS suggests more frequent visual updates, which can correlate with content-rich presentations such as lecture slides or information-heavy visuals. While this method does not perform direct text extraction, it provides a practical and scalable estimate of the video's textual density. The resulting score is scaled to fall within the range [0, 100].

The obtained scores were normalized so that the visual, auditory and textual scores, represented respectively by v_c, a_c and t_c add up to 5 as mentioned in section 4.2. The code used to obtain the above scores can be found in the GitHub link to this Colab notebook: `https://github.com/Serena-Anthony/vat-score-calculation`

For **textual materials**, since they inherently lack visual and audio elements, the VAT score was simplified to focus solely on the textual component. In this case, the visual and audio scores were set to zero. The textual score was set to 5.

This approach allowed us to consistently measure and compare the perceived quality and engagement level across both video and text formats using a unified scoring framework.

Information Processing

To evaluate a student's information processing ability for each Class 6 Science topic, we use **Bloom's Taxonomy**. Bloom's taxonomy categorizes cognitive skills into six levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. Each level reflects a deeper or more complex understanding. Unlike the proof of concept, ip score is associated with a **topic** instead of LO. For each topic, a score between 0 and 1 is assigned at each level based on how well the topic allows or encourages students to engage with that cognitive skill.

For example, a topic like ”Balanced Diet” can support higher-order thinking such as creating personalized meal plans or evaluating dietary choices, so it scores higher in the Create and Evaluate levels. In contrast, a topic like ”Food Intro” might mainly involve recalling facts, so it scores higher in Remember but lower in advanced levels. The final IP score is the sum of all six levels, with a maximum score of 6, representing the topic’s overall cognitive engagement potential. Table 5.12 shows the IP scores assigned to each topic in the course after consulting grade 6 Science teachers from nearby schools.

Average Difficulty

To assign average difficulty values for the topics, grade 6 science teachers were approached to obtain their general consensus. With their results, difficulty scores were assigned for each topic in the range [0,5]. Table 5.13 shows the difficulty scores assigned to each topic in the course.

Topic	**Remember**	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Create**	**Final IP Score**
Food	1.0	0.8	0.4	0.3	0.2	0.1	2.8
Food Sources	1.0	1.0	0.6	0.5	0.3	0.2	3.6
Plant Sources	1.0	0.9	0.7	0.5	0.3	0.2	3.6
Animal Sources	1.0	0.9	0.6	0.4	0.2	0.1	3.2
What do Animals Eat?	0.8	0.9	0.7	0.7	0.4	0.3	3.8
Components of Food	1.0	1.0	0.8	0.6	0.5	0.3	4.2
Nutrients	1.0	1.0	0.9	0.6	0.5	0.4	4.4
Deficiency Disease	1.0	1.0	0.8	0.7	0.6	0.4	4.5
Balanced Diet	0.8	1.0	1.0	1.0	0.7	0.5	5.0

Table 5.12: IP Scores across different topics

5.2.5 Learning pathway generation module

Once the profiling assessment has been completed by the student, a personalized learning path is generated using the learning objects stored in the `lo` table. The algorithm utilizes the user profile from the `mdl_user` table and the metadata of learning objects from the `lo` table to create a customized path. Each learning object is represented by its corresponding `id` from the `lo` table, and a chromosome is constructed as a sequence of these `id`s. The path generation process follows the methodology outlined in Section 4.3.8, and the implementation

Topic	Difficulty Score
Food	1.2
Food Sources	2.0
Plant Sources	1.5
Animal Sources	2.0
What do Animals Eat?	3.5
Components of Food	4.5
Nutrients	4.5
Deficiency Disease	3.5
Balanced Diet	5.0

Table 5.13: Difficulty Scores for Each Topic

is adapted directly from the proof of concept. Each learning object is then rendered to the student based on its `id`. The sequence of file calls for this process is illustrated in Fig. 5.12, while a generalized overview of all file calls in the system is presented in Fig. 5.13.

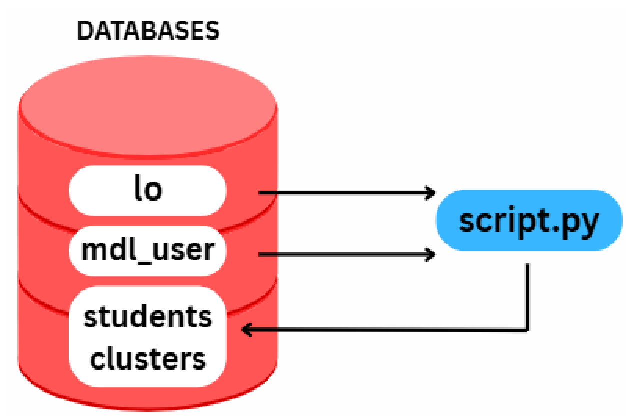

Figure 5.12: Learning pathway generation

5.2.6 Optimization-clustering

The student profiles are clustered on the basis of v,a,t scores as mentioned in Section 4.4. Adding students to clusters is handled by `script.py`. The initial population of chromosomes

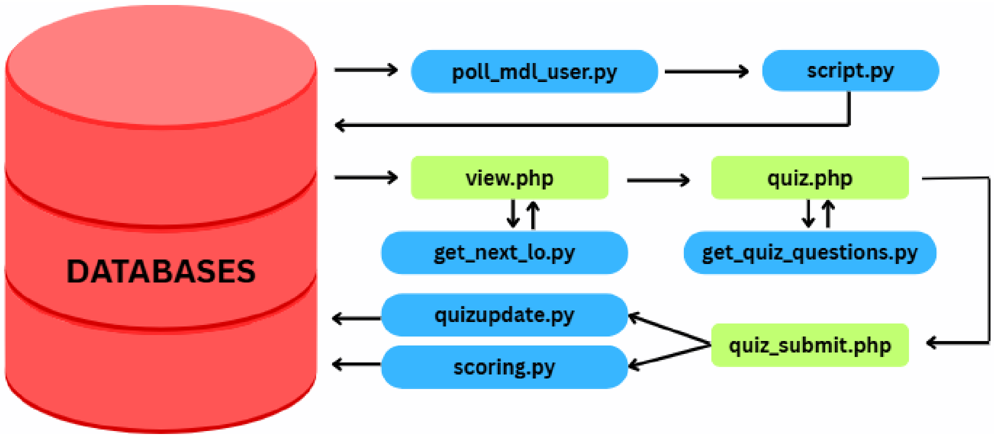

Figure 5.13: Overview of the file calls of the system

is determined by the cluster that the student gets added to. A personalized path is then evolved from this initial population.

When a student is added to an existing cluster, the initial population is generated from the gene pool of the cluster. This results in faster convergence. Hence, the algorithm is run for fewer generations. If a new cluster is assigned to a student, on the other hand, the GA must be run for a higher number of generations.

To increase the efficiency of the GA, student profiles of 75 students from grade 6 and 7 were gathered through the profiling assessment. The path generation algorithm was run for these students, thus creating a set of clusters and gene pool. This increases the chances of a new student belonging to an existing cluster, making the path generation process more efficient.

5.2.7 Course Pathway

The path generated above is rendered to the student using `view.php`. The file obtains the learning object to be displayed using `get_next_lo.py`. Once the lo is completed, the student is redirected to a quiz on the completed topic. The file calls for the same are shown in Fig. 5.14. Fig. 5.15 shows how the learning objects are displayed to the students.

5.2.8 Evaluation module

Pre-assessment

The pre-assessment is conducted to evaluate the students prior knowledge and readiness for the course. It comprises a set of 10 MCQs covering a range of topics selected from the chapters we had taken for this project. Fig. 5.16 illustrates the pre-assessment interface as

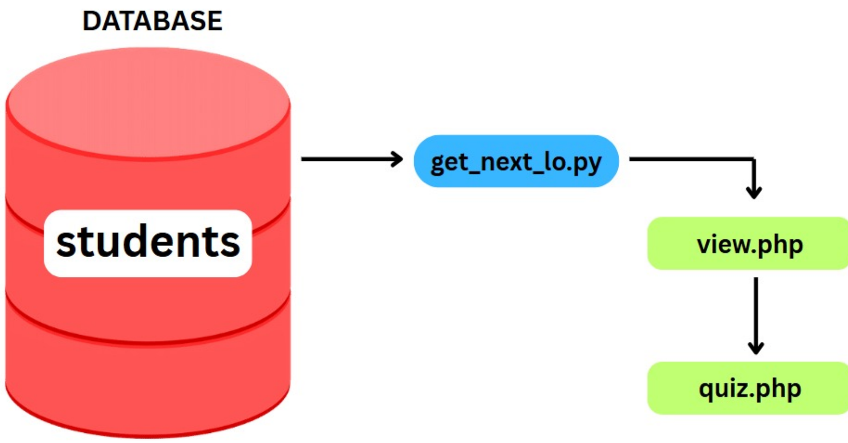

Figure 5.14: File calls to display learning object

Figure 5.15: Displaying learning object

implemented in the Moodle platform.

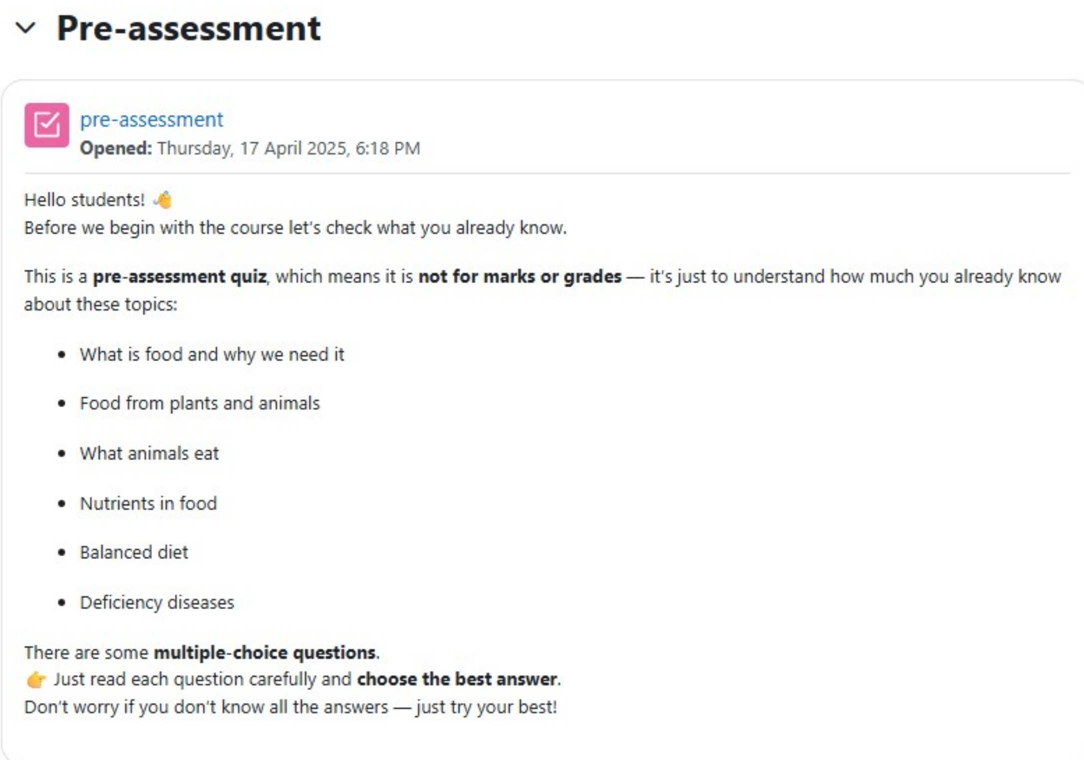

Figure 5.16: Pre-assessment interface

Post assessment

Post assessments are conducted continuously throughout the course. After each lesson a post assessment is conducted to determine whether the student has understood the topic. If the student successfully passes the assessment, they progress to the next topic in the sequence. However, if they do not pass, they are given more resources on the same topic. This process continues until the student demonstrates understanding or all available materials from the content database have been exhausted.

Summative Assessment

The summative assessment is conducted at the end of the course, after the student has completed the personalised course path. It comprises a set of 10 MCQs covering a range of topics addressed throughout the course. This assessment is intended to measure the extent of the student's learning and knowledge gained after the course. Fig. 5.17 illustrates the summative assessment interface as implemented in the Moodle platform.

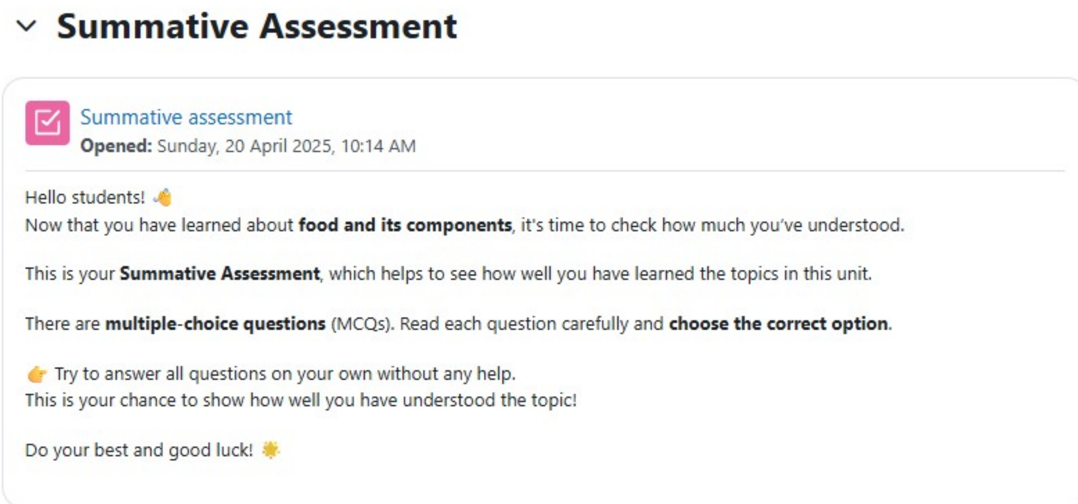

Figure 5.17: Summative assessment interface

Question Bank

We created three sets of question banks on moodle. The first set is for the pre-assessment. It includes easy-to-moderate-level questions designed to assess students' initial understanding of the subject. The second set was created for the summative assessment. It includes a diverse mix of questions aimed at measuring the knowledge that the student has gained after completing the course with a personalized learning path. The final question bank consisting of topic-wise questions, which is used to ask a set of post-lesson questions. For each topic a varied number of questions is asked in the post-assessment and the number of questions is specified in the table `topics`. These post-assessment questions help determine whether the student has sufficiently understood the topic or if additional instructional content should be provided before progressing further.

Rendering quiz

Pre-assessment and summative assessments are rendered as moodle quizzes, with the questions from corresponding question banks. Post-assessments are rendered by `quiz.php`. After each lo, `view.php` redirects the student to `quiz.php` for a quiz on the previously taught topic. The quiz is displayed as a form. Quiz questions are obtained directly from the database through `get_quiz_questions.py` . When the quiz is submitted, `quiz_submit.php` is invoked. It calculates the quiz score and displays it to the student. Additionally, the solutions to the attempted questions are also made visible, with appropriate feedback. It then executes `scoring.py` to store the quiz score in `quiz_attempts` and `Quizupdate.py` to update the learning path of the student. The file calls are as shown in Fig. 5.18. Fig. 5.19 and Fig. 5.20 show the quiz and the quiz submission pages respectively.

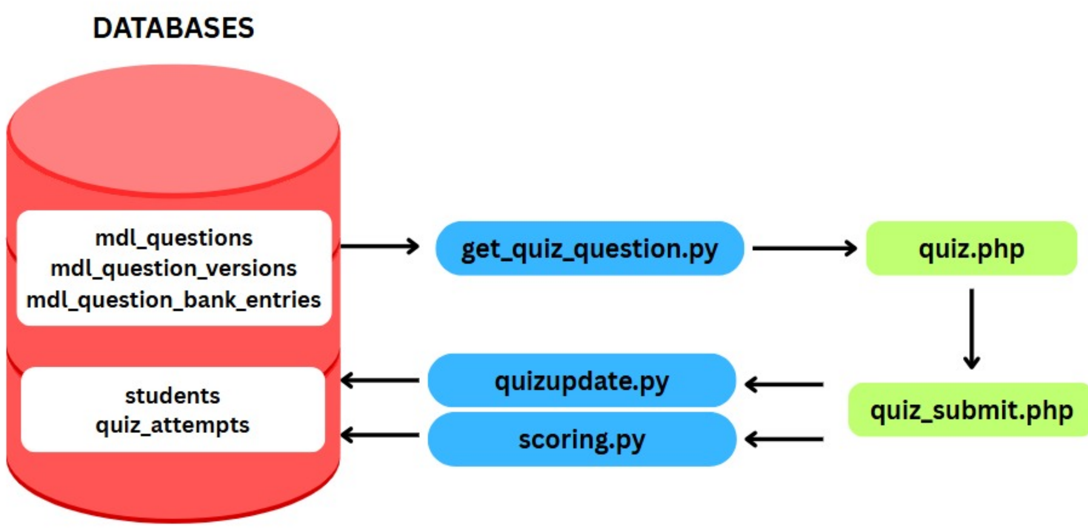

Figure 5.18: Rendering post assessment

5.2.9 Learning style dashboard

Once the profiling assessment is complete, the students may view a dashboard showing their learning style scores as a bar graph. It invokes `get_dashboard_data.py` to obtain the scores of the student from the database. The file calls are as shown in Fig. 5.21. The dashboard is shown in Fig. 5.22.

5.2.10 Learning progress dashboard

As the student progresses through the course, their quiz scores are stored in the `quiz_attempts`. These scores are obtained by `get_progress_data.py` to display a dashboard indicating the topics completed by the student and the scores obtained in the corresponding attempts. The file calls are as shown in Fig. 5.23. The dashboard is shown in Fig. 5.24.

5.3 Conclusion

This section detailed the prototype implementation of the proposed framework. A functional code base was developed from the proof of concept for learning path generation and subsequently integrated with Moodle, the chosen Learning Management System (LMS). A

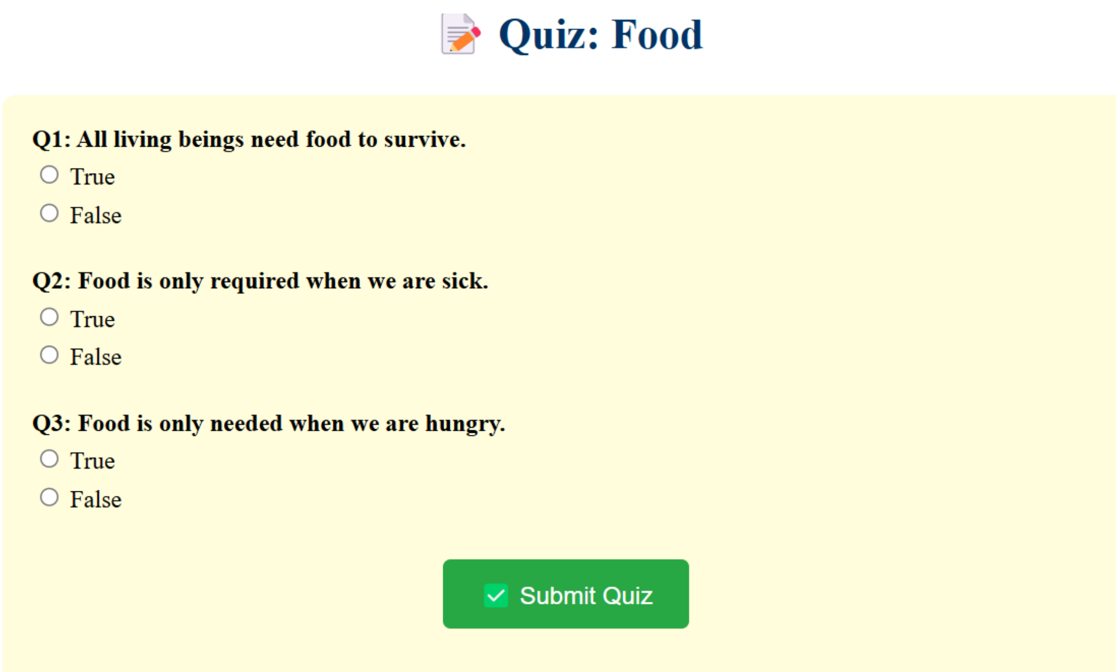

Figure 5.19: Quiz page

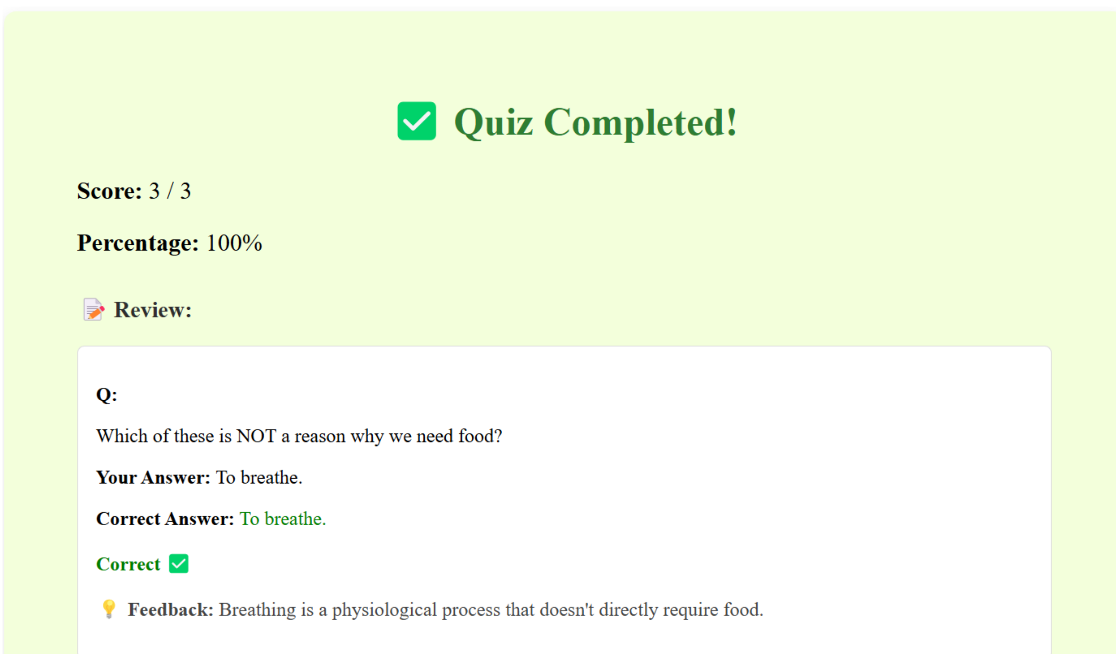

Figure 5.20: Quiz submission page

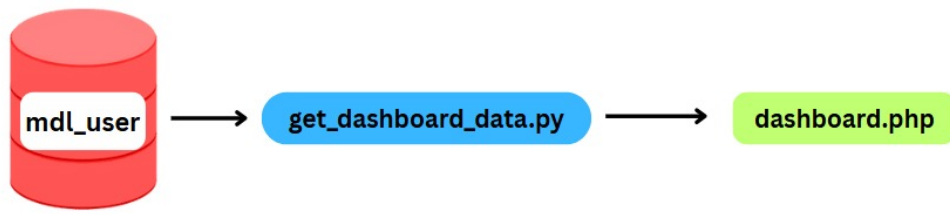

Figure 5.21: Rendering learning style dashboard

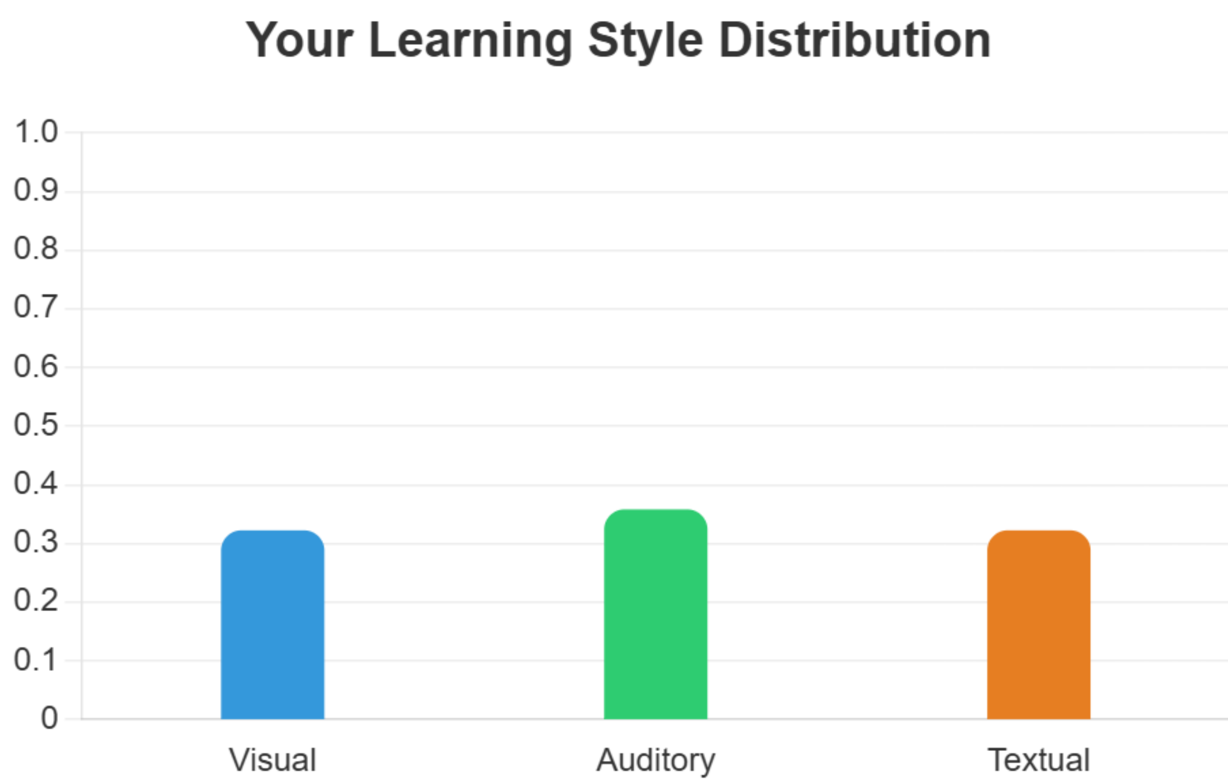

Figure 5.22: Learning style dashboard

Figure 5.23: Rendering learning progress dashboard

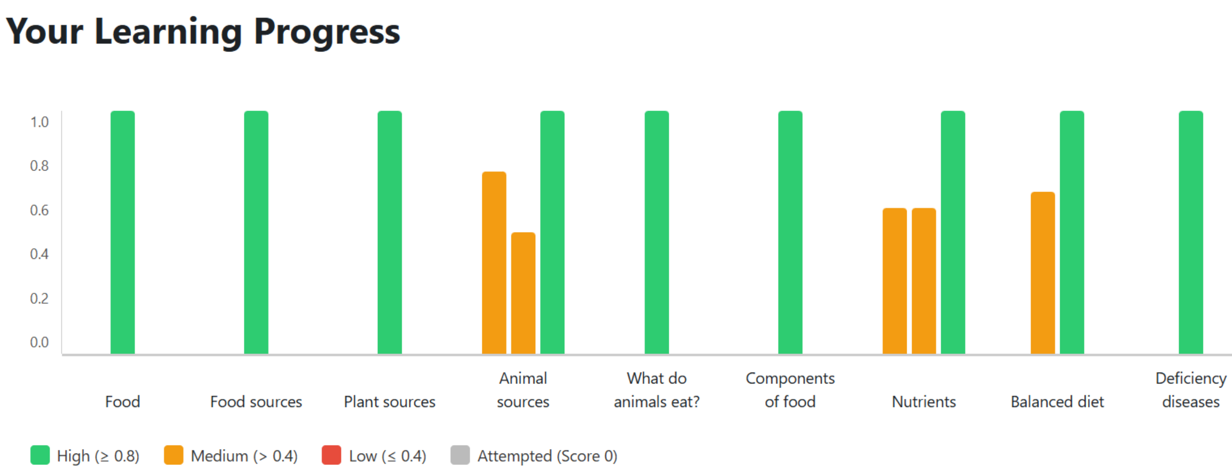

Figure 5.24: Learning progress dashboard

web interface was created to collect user profiles through cognitive assessments. Instructional content was sourced from NCERT Grade 6 science chapters, focusing on videos and textual materials. These learning objects were annotated with relevant metadata, including modality, readability, difficulty, and cognitive demands. Pre-assessments and summative assessments were implemented as Moodle quizzes, while post-assessments were conducted using HTML forms after each learning object to dynamically update the learning path. Together, these components form a comprehensive system for adaptive learning, as discussed so far. The next chapter presents the experiments conducted and the results obtained from them.

Chapter 6

Experiments and Results

6.1 USER PROFILING

As part of the data collection process for clustering, profiles of 75 students from nearby schools were collected. To validate the reliability and relevance of the cognitive assessments used in profiling, a subset of 6 students was selected based on their assessment scores. Their academic performance was subsequently reviewed in consultation with their respective school faculty. Table 6.1 summarizes the assessment scores and the academic categorization provided by the faculty for each of these students.

Table 6.1: Assessment Scores and Faculty Categorization of Selected Students

Student	**Flesch score**	**ipv**	**ipa**	**ipt**	**wmv**	**wma**	**wmt**	**Category**
1	64	2	2	3	1	3	1	Below Average
2	96	6	5	6	4	5	5	Good
3	100	4	1	5	2	2	4	Above Average
4	88	4	5	6	5	5	5	Good
5	82	6	5	6	5	4	5	Excellent
6	54	6	6	6	5	4	5	Good

The student profiling assessment scores align with the general review provided by the respective faculties, thereby validating the effectiveness of the profiling mechanism.

S.No	Selection	Mutation Probability	Best Fitness			Average Fitness			Norm. Consistency	Norm. Convergence
			Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3		
1	Tournament	0.3	-10.014	-8.575	-10.013	-10.060	-8.613	-10.015	0.000000	0.00
2	Tournament	0.4	-10.014	-8.679	-10.013	-10.016	-10.014	-10.017	0.883598	0.02
3	Tournament	0.5	-10.014	-8.450	-10.013	-10.167	-8.487	-10.014	0.894180	0.02
4	Tournament	0.7	-10.014	-6.0679	-10.013	-10.016	-6.375	-10.022	1.000000	0.25
5	Tournament	0.8	-10.014	-6.0679	-10.013	-10.024	-6.665	-10.021	1.000000	0.25
6	Rank	0.3	-8.299	-6.0679	-8.665	-14.870	-9.108	-14.957	0.932981	0.54
7	Rank	0.4	-6.0679	-8.722	-6.0679	-9.483	-16.357	-11.110	0.941799	0.75
8	Rank	0.5	-6.0679	-7.922	-6.0679	-9.868	-13.572	-10.097	0.906526	0.82
9	Rank	0.7	-6.0679	-7.719	-6.0679	-11.643	-12.738	-9.268	0.897707	0.84
10	Rank	0.8	-6.0679	-7.748	-6.0679	-8.319	-12.112	-6.657	0.899471	0.84
11	Roulette Wheel	0.3	-6.0679	-7.902	-6.0679	-11.173	-12.145	-7.439	0.904762	0.83
12	Roulette Wheel	0.4	-6.0679	-6.0679	-6.0679	-7.322	-11.436	-7.243	0.823633	1.00
13	Roulette Wheel	0.5	-6.0679	-6.0679	-7.748	-10.087	-7.684	-11.723	0.899471	0.84
14	Roulette Wheel	0.7	-6.0679	-6.0679	-6.0679	-6.760	-6.979	-7.210	0.823633	1.00
15	Roulette Wheel	0.8	-6.0679	-6.0679	-6.0679	-10.798	-10.561	-6.691	0.823633	1.00

Table 6.2: Comparison of selection methods, mutation probabilities, and fitness values including Normalized Consistency and Convergence

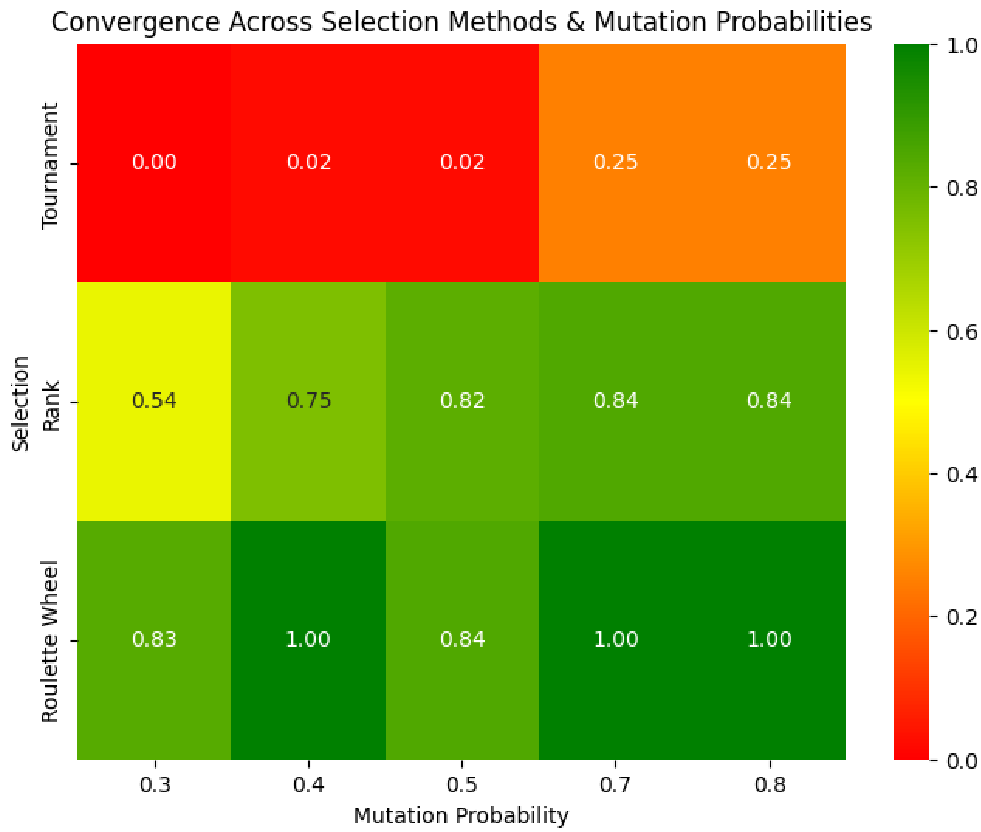

Figure 6.1: Heatmap - Convergence

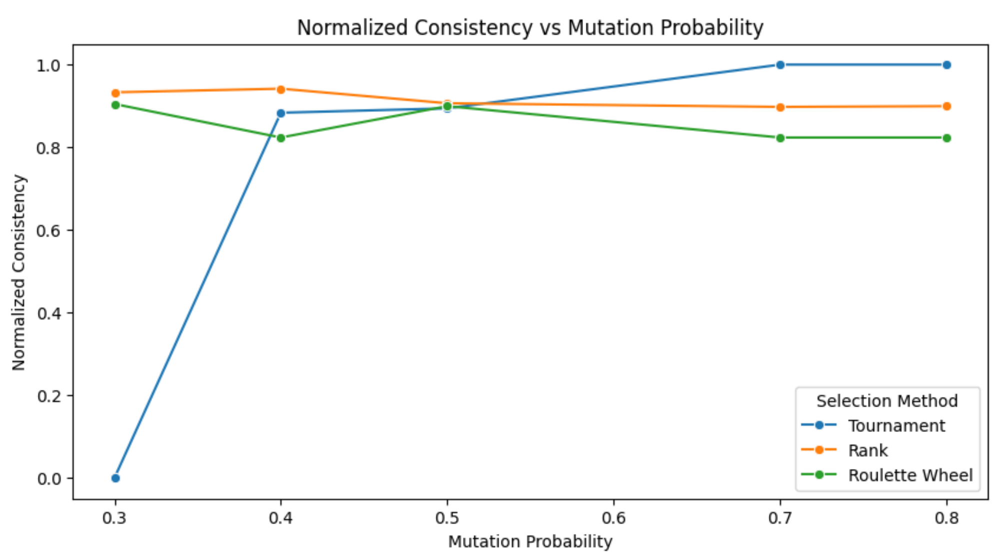

Figure 6.2: Line plot - Consistency

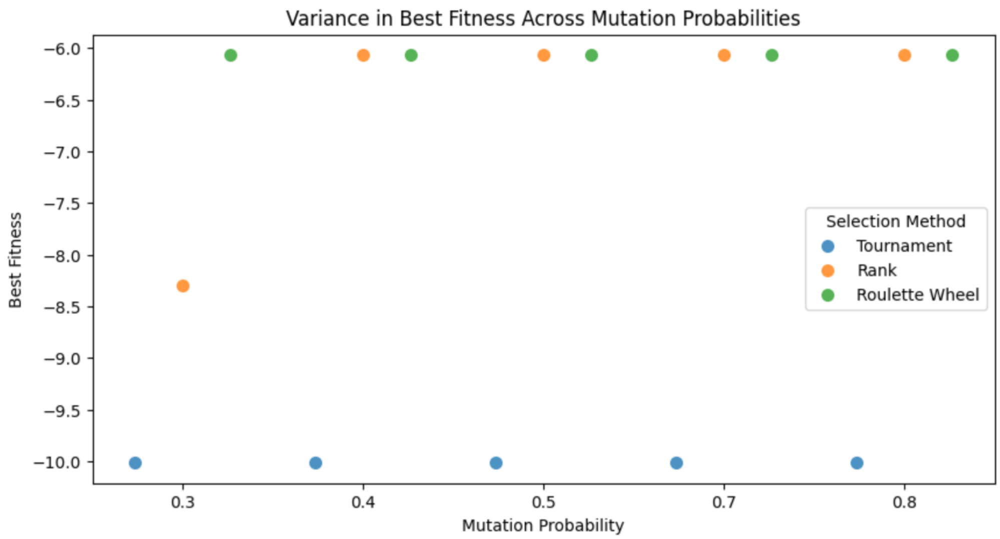

Figure 6.3: Swarm plot - Variance

6.2 PARAMETER SELECTION

Parameters such as the selection strategy and mutation probability were empirically determined through observations derived from the proof-of-concept implementation.

An experiment was carried out using the proof of concept algorithm, wherein three selection strategies—tournament, rank-based, and roulette wheel—were evaluated across five mutation probabilities: 0.3, 0.4, 0.5, 0.7, and 0.8. For each selection–mutation configuration, three independent trials were conducted to record the best fitness and average fitness values. Based on the best fitness values obtained across these trials, consistency and convergence metrics were subsequently computed to evaluate the stability and effectiveness of each configuration.

The values obtained were generated using synthetic data consisting of randomly initialized student profiles and instructional content. Table 6.2 presents the resulting values corresponding to each configuration. These results are intended solely for illustrative purposes and serve to validate the design and functionality of the proposed framework.

Normalized Consistency: Normalized Consistency measures how stable the best fitness values are across multiple trials. If f_1, f_2, f_3 represent the best fitness values obtained in three independent trials, then the average best fitness is calculated as:

$$\bar{f} = \frac{f_1 + f_2 + f_3}{3}$$

The standard deviation of these best fitness values is:

$$\texttt{std\_dev\_best\_fitness} = \sqrt{\frac{(f_1 - \bar{f})^2 + (f_2 - \bar{f})^2 + (f_3 - \bar{f})^2}{3}}$$

To obtain the normalized consistency score, the standard deviation is scaled relative to the maximum observed deviation across all configurations:

$$\text{Normalized Consistency} = 1 - \frac{\texttt{std\_dev\_best\_fitness}}{\max(\texttt{std\_dev\_best\_fitness})}$$

Normalized Convergence: Normalized Convergence quantifies how closely the average fitness across trials approaches the global optimum. If f_1, f_2, f_3 denote the best fitness values from three trials, the average fitness is:

$$\bar{f} = \frac{f_1 + f_2 + f_3}{3}$$

Then, the average absolute deviation from this mean is given by:

$$\texttt{deviation} = \frac{1}{3}\left(|f_1 - \bar{f}| + |f_2 - \bar{f}| + |f_3 - \bar{f}|\right)$$

The normalized convergence is calculated by scaling this deviation relative to the maximum deviation observed:

$$\text{Normalized Convergence} = 1 - \frac{\texttt{deviation}}{\max(\texttt{deviation})}$$

The tabulated values were subsequently visualized using three different plots to aid in result interpretation:

1. A heatmap illustrating the convergence values across the three selection strategies and five mutation probabilities.

2. Line plot showing the consistency values observed across the same configurations.

3. A swarm plot showing the distribution and variance in the best fitness values for each combination of selection strategy and mutation probability.

These visualizations help in identifying patterns and assessing the impact of different genetic algorithm parameters on performance metrics.

6.2.1 Selection Strategy

Roulette wheel selection was chosen as the preferred selection strategy, as it demonstrated superior convergence and lower variance while maintaining competitive consistency compared to tournament and rank-based selection methods, as illustrated in Fig. 6.1, Fig. 6.2, and Fig. 6.3.

6.2.2 Mutation Probability

Based on the results Table. 6.2 and heatmap in Fig. 6.1, the mutation probability is set to **0.7**. It offers the highest average fitness values for the three trials of roulette wheel selection and high convergence.

6.2.3 Algorithm Execution

- The genetic algorithm is run for **1000 generations** during the initial learning path generation phase. This extended run allows the algorithm to explore a broader range of possible learning paths and effectively converge towards an optimal or near-optimal solution, particularly when starting from a cold profile without prior learning history.

- The genetic algorithm is run for **500 generations** during the path regeneration phase after failed post-assessment. Since this stage builds on an already guided path and benefits from previous evaluation, fewer generations are sufficient to refine the learning trajectory based on updated performance data.

6.2.4 Student Profile Clustering

- **Manhattan distance** is used to measure the distances between student profiles and cluster centroids. This distance metric is suitable for ordinal data as it considers the absolute differences across all profile features.

- The *threshold distance* is set to **0.1**, based on experiments conducted not only with the initial proof of concept but also on the prototype using properly structured instructional content and metadata. This value allows for the formation of a sufficient number of meaningful clusters. Distances greater than 0.1 were found to result in very few clusters due to the overall proximity among student profiles, which limited the algorithm's ability to distinguish between subtle differences in learner characteristics.

$$D_{\text{Manhattan}}(p, q) = \sum_{i=1}^{n} |p_i - q_i| \tag{6.1}$$

6.3 FITNESS FUNCTION

Experiments were conducted on the prototype system using two different formulations of the fitness function:

- **Fitness Function 1 (Complete)**: This function f incorporates both the mean learning cost and the variance across gene-level learning costs, as defined in Equation 4.9.
- **Fitness Function 2 (Simplified)**: This function considers only the mean learning cost lc, as defined in Equation 4.7.

To evaluate the effectiveness of these two formulations, two synthetic student profiles were created—one with a strong preference for visual learning and another with a preference for textual learning. The genetic algorithm was executed for each profile using both fitness functions, and the resulting learning paths were compared in terms of the types of learning objects selected.

The results are summarized in Table 6.3:

Table 6.3: Comparison of Learning Object Types in Generated Paths

Student Profile	Fitness Function	Text Objects	Video Objects
Visual Learner	Fitness 1 (Complete)	3	6
Visual Learner	Fitness 2 (Simplified)	3	6
Textual Learner	Fitness 1 (Complete)	7	2
Textual Learner	Fitness 2 (Simplified)	8	1

From the table, it can be observed that for the visual learner, both fitness functions produced similar results. However, for the textual learner, Fitness Function 2 (Simplified) produced a path with a higher number of textual learning objects and fewer videos, better aligning with the learner's preferences.

This suggests that the simplified fitness function based solely on the mean learning cost may provide better personalization in cases with distinct learning style preferences. Additionally, since Fitness Function 2 eliminates the variance computation, it reduces the overall computational complexity of the genetic algorithm, resulting in faster convergence and lower processing overhead—an advantage particularly relevant for real-time or large-scale adaptive learning systems.

6.4 COURSE GENERATION

Three students were enrolled in the course page and proceeded to complete the course workflow. Each student began by taking the mandatory profiling assessment, followed by a pre-assessment to establish a baseline of their initial understanding.

As outlined in Section 4.4, clusters were created based on the profiling data collected from 75 students. A new student was either assigned to the nearest cluster if the computed distance was below a predefined threshold, or a new cluster was created if the threshold was exceeded. Personalized learning paths were then generated by selecting genes from the gene pool associated with the respective cluster.

The students navigated through the personalized learning paths generated for them, with a post-assessment quiz administered after each learning object to evaluate knowledge acquisition.

Upon completion of the entire learning path, the students undertook a summative assessment to measure overall learning gains and content mastery.

Table 6.4 presents the pre-assessment, post-assessment, and summative assessment scores for the three students. It is important to note that the average difficulty level of the summative assessment questions was higher than that of the pre-assessment.

Student	Pre-assessment	Summative assessment
1	0.8	0.8
2	1.0	0.9
3	1.0	1.0

Table 6.4: Pre-assessment and summative assessment scores for the three students

The post assessment scores of students 1,2 and 3 are indicated by the learning progress dashboard of the student in Fig. 6.4, Fig. 6.5 and Fig. 6.6 respectively.

Additionally, two dummy profiles were created to understand the performance of the algorithm. A profile was created such that the student is completely a visual learner and another such that the student is completely a textual learner. The learning paths generated for the above profiles were analyzed. Note that there are a total of 9 topics in the course graph. Besides, the learning objects gathered include only videos and text, no audio content exclusively. The findings are as summarized below:

- The visual learner received 6 video and 3 textual resources.
- The textual learner received 1 video and 8 textual resources.
- The fitness values of the chromosomes are influenced by information processing and readability scores in addition to the learning style scores. This justifies why an all-textual or all-visual content is not generated for the above learners.

These findings demonstrate the system's ability to generate learning paths that meaningfully align with learner preferences while still balancing other cognitive constraints, validating the effectiveness of the proposed adaptive framework.

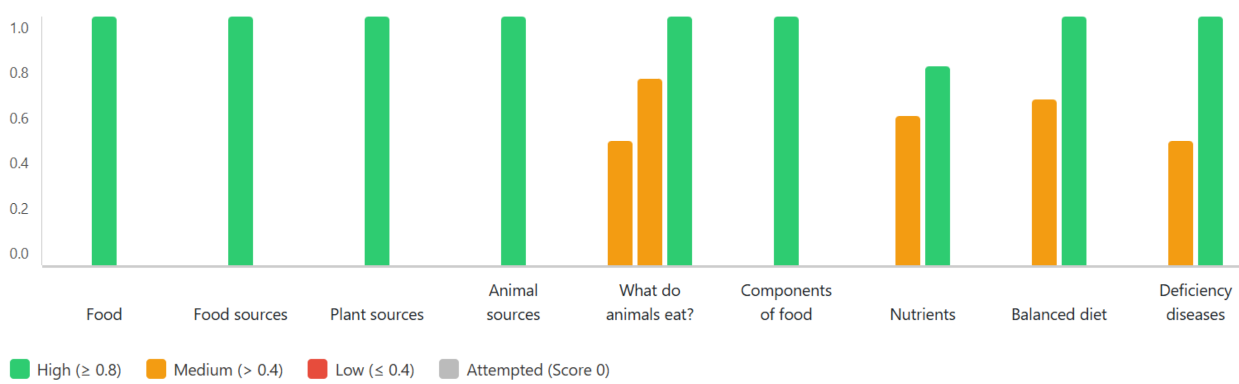

Figure 6.4: Post assessment scores of student 1

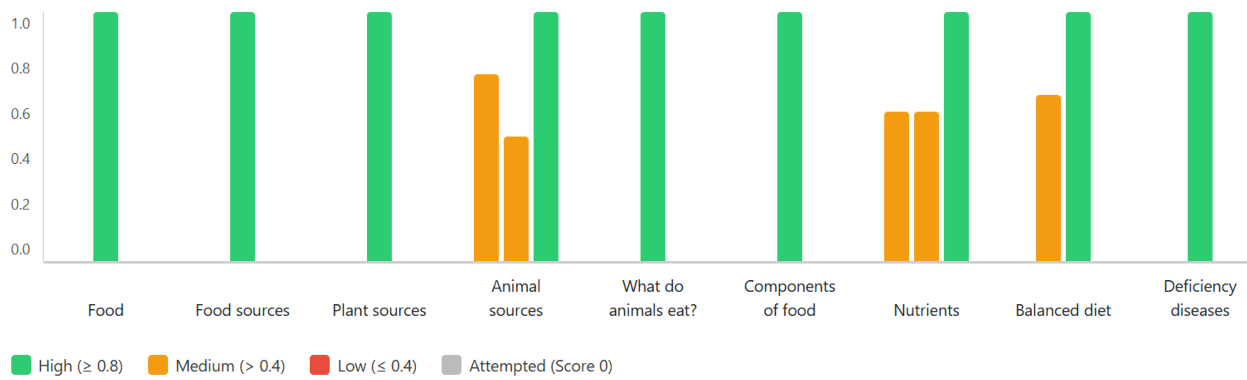

Figure 6.5: Post assessment scores of student 2

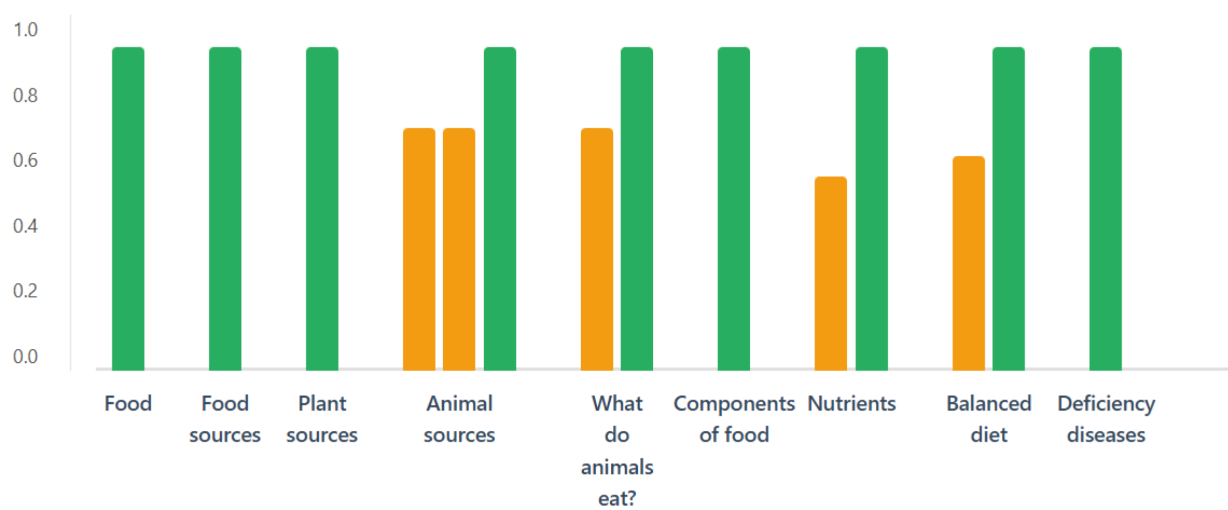

Figure 6.6: Post assessment scores of student 3

Chapter 7

Conclusion

This project set out to design and implement an adaptive e-learning framework that personalizes content delivery based on a learner's cognitive profile, with a specific focus on inclusivity for neurodiverse students. By harnessing the optimization capabilities of Genetic Algorithms (GAs), the system dynamically constructs and refines learning paths that evolve in response to student performance and interaction over time. It consists of four main modules:

- **User Profile Module:** Collects data on each learner's cognitive attributes, including working memory capacity, information processing ability, and content readability preference, via standardized assessments.
- **Instructional Content Module:** Stores learning resources along with rich metadata (e.g., modality, difficulty, readability), enabling precise matching to student needs.
- **Learning Pathway Generation Module:** Uses a Genetic Algorithm to generate personalized learning paths by optimizing content sequences according to the learner's profile and ongoing performance.
- **Evaluation Module:** Monitors quiz performance and learner interaction to update fitness values, triggering dynamic adaptations in the learning path, including topic substitutions when students face persistent difficulty.

The framework was fully implemented on the Moodle platform using actual instructional content and metadata, enabling real-world deployment and validation. Controlled experiments were conducted using synthetic profiles—a highly visual learner and a highly textual learner—to evaluate the system's ability to adapt learning paths in accordance with individual styles. While the broader aim of fully addressing neurodiversity remains an ongoing challenge, the framework successfully demonstrated real-time adaptation and personalization across diverse learning preferences.

The key contributions of this work are as follows:

1. **Broader personalization:** The system effectively adapts content delivery to diverse cognitive and learning style preferences, including visual and textual modalities.

2. **Novel use of Genetic Algorithms:** GAs, while not rare, remain underexplored in adaptive learning. In this work, they were used to evolve learning paths based on a multi-factorial fitness function incorporating difficulty, readability, learning style compatibility, and learner feedback—all without requiring large datasets or complex reward shaping.

3. **Fully implemented and validated:** Unlike many theoretical frameworks, this solution was deployed on Moodle and tested with real content, proving both technical viability and scalability.

4. **In-course dynamic adaptation:** The learning paths are adjusted in real-time based on quiz performance, allowing the system to respond to evolving learner needs. When a student consistently struggles with a topic, the framework intelligently interchanges it with an alternative topic of similar difficulty, helping them overcome learning roadblocks while maintaining pedagogical continuity.

5. **Low-data compatibility:** The GA framework performs well in data-sparse environments, making it suitable for real-world educational settings lacking large training corpora.

6. **Maintained instructional continuity:** The algorithm ensures logical flow between learning objects, a commonly overlooked factor in adaptive systems.

In conclusion, this project successfully meets its core objective of providing a personalized and adaptive learning experience by leveraging Genetic Algorithms (GAs) to tailor educational content based on learner performance and preferences. By addressing existing limitations in adaptive learning systems and validating the solution in a real-world platform, our work offers a meaningful foundation for future research and deployment of more personalized and inclusive educational technologies.

Chapter 8

Future Work

Future work related to this project may majorly focus on the below mentioned areas:

- The project was initiated with an aim to create a framework for neurodiverse students. Given the time constraints, this may be far-fetched. As the GA evolves personalized content, we are now one step closer to creating a platform that is user-friendly for neurodiverse students too. The project may thus be extended in this direction.

- The metrics used for student profiling and instructional content can be further broadened. The metrics can be refined to better reflect the abilities of the students. The instructional content metadata can be extracted more efficiently, encompassing various features of the content.

- The learning objects can be extended to include more varied content. Currently, videos and text materials are the resources delivered to the students. This can further include audio books, interactive images among others to expand the variety of content.

- Gamification elements can be introduced to enhance student engagement and motivation. Features such as badges, leaderboards, and interactive challenges can be integrated into the learning environment to make the experience more immersive and rewarding, especially for students who benefit from a more dynamic and visually stimulating learning approach.

Chapter 9

Appendix

9.1 INSTRUCTIONAL CONTENT CREDITS

This section includes the websites and materials that were used as instructional content in our prototype.

9.1.1 Video content references

- https://www.youtube.com/watch?v=qhbdqKUwk0w
- https://www.youtube.com/watch?v=cmupZ8DuB3I
- https://www.youtube.com/watch?v=AKKnhhFO50k
- https://www.youtube.com/watch?v=44MsQw8VjEs&t=8s
- https://www.youtube.com/watch?v=wi59b8fX0AE
- https://www.youtube.com/watch?v=d0aUIesWWB8
- https://www.youtube.com/watch?v=e0kkwDxQJRs&t=236s
- https://www.youtube.com/watch?v=Lp7VBh2gx38
- https://www.youtube.com/watch?v=MYUU8ZcGGt8
- https://www.youtube.com/watch?v=rJw4RpFRhBo
- https://www.youtube.com/watch?v=Qt7ZCc4JlzA
- https://www.youtube.com/watch?v=7YPp4eMf6lQ

- https://www.youtube.com/watch?v=dUFGB-WWw88
- https://www.youtube.com/watch?v=5aLBa46yCIQ&t=897s

9.1.2 Textual content references

- https://byjus.com/cbse-notes/cbse-class-6-science-notes-chapter-1-food-where-does-
- https://www.nextgurukul.in/wiki/concept/cbse/class-6/science/food-where-does-it-com sources-of-food/3957756
- https://www.vedantu.com/biology/food-source
- https://byjus.com/cbse/food-variety-and-sources/
- https://byjus.com/biology/food-sources-animal-plant-products/
- https://byjus.com/cbse/components-of-food/
- https://www.vedantu.com/biology/components-of-food
- https://byjus.com/biology/nutrients/
- https://classnotes.org.in/class-6/science/components-of-food/balanced-diet/ #:~:text=A%20balanced%20diet%20should%20contain%20contain%20enough%20food, and%20fruits%20%28vitamins%2C%20minerals%20and%20roughage%29%205%20Water
- https://byjus.com/biology/food-deficiency/
- https://www.vedantu.com/biology/food-deficiency
- https://byjus.com/biology/food-sources-animal-plant-products/
- https://byjus.com/biology/carnivores-herbivores/
- https://byjus.com/cbse-notes/cbse-class-6-science-notes-chapter-1-food-where-does- #what-do-animals-eat
- https://allen.in/cbse-notes/class-6-science-notes-chapter-1-components-of-food
- https://www.learncbse.in/food-come-cbse-notes-class-6-science/
- https://www.learncbse.in/components-food-cbse-notes-class-6-science/

9.1.3 Tools used

- Flesch reading score calculator
  - https://serpninja.io/tools/flesch-kincaid-calculator/#:~:text=The%20Flesch%20Reading%20Ease%20Score%20ranges%20from%200%20to%20100,complex%20and%20difficult%20to%20understand
- Online transcribers
  - https://vizard.ai/tools/video-to-text
  - https://turboscribe.ai/dashboard

References

[1] A.-L. Le Cunff, E. Dommett, and V. Giampietro, "Supporting neurodiversity in online education: a systematic review," *McKenzie SP, Arulkadacham L, Chung J, Aziz Z, Herausgeber. The future of online education. Nova Publishers*, pp. 213–34, 2022.

[2] S. R. Ross, "Supporting your neurodiverse student population with the universal design for learning (udl) framework," in *2019 IEEE Frontiers in Education Conference (FIE)*, pp. 1–5, IEEE, 2019.

[3] Y. Kritikou, P. Demestichas, E. Adamopoulou, K. Demestichas, M. Theologou, and M. Paradia, "User profile modeling in the context of web-based learning management systems," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 603–627, 2008.

[4] M. M. Alhawiti and Y. Abdelhamid, "A personalized e-learning framework.," *Journal of education and e-learning research*, vol. 4, no. 1, pp. 15–21, 2017.

[5] I. Gligorea, M. Cioca, R. Oancea, A.-T. Gorski, H. Gorski, and P. Tudorache, "Adaptive learning using artificial intelligence in e-learning: a literature review," *Education Sciences*, vol. 13, no. 12, p. 1216, 2023.

[6] S. Amin, M. I. Uddin, A. A. Alarood, W. K. Mashwani, A. Alzahrani, and A. O. Alzahrani, "Smart e-learning framework for personalized adaptive learning and sequential path recommendations using reinforcement learning," *IEEE Access*, 2023.

[7] W. S. Sayed, A. M. Noeman, A. Abdellatif, M. Abdelrazek, M. G. Badawy, A. Hamed, and S. El-Tantawy, "Ai-based adaptive personalized content presentation and exercises navigation for an effective and engaging e-learning platform," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3303–3333, 2023.

[8] X.-h. Tan, R.-m. Shen, and Y. Wang, "Personalized course generation and evolution based on genetic algorithms," *Journal of Zhejiang University SCIENCE C*, vol. 13, no. 12, pp. 909–917, 2012.

[9] M.-J. Huang, H.-S. Huang, and M.-Y. Chen, "Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach," *Expert Systems with applications*, vol. 33, no. 3, pp. 551–564, 2007.

[10] O. C. Agbonifo and O. A. Obolo, "Genetic algorithm-based curriculum sequencing model for personalised e-learning system," *International Journal of Education and Management Engineering*, vol. 5, no. 8, pp. 27–35, 2018.

[11] K. Maycock, *A framework for adaptive e-learning.* PhD thesis, National University of Ireland Maynooth, 2010.

[12] I. Alvarez-Icaza, J. Molina-Espinosa, and P. Suárez-Brto, "Adaptive learning for complex thinking: A systematic review of users' profiling strategies," *Journal of Social Studies Education Research*, vol. 15, no. 2, pp. 251–272, 2024.

[13] S. M. Jaeggi, M. Buschkuehl, W. J. Perrig, and B. Meier, "The concurrent validity of the n-back task as a working memory measure," *Memory*, vol. 18, no. 4, pp. 394–412, 2010.

[14] S. Pelegrina, M. T. Lechuga, J. A. García-Madruga, M. R. Elosúa, P. Macizo, M. Carreiras, L. J. Fuentes, and M. T. Bajo, "Normative data on the n-back task for children and young adolescents," *Frontiers in psychology*, vol. 6, p. 1544, 2015.

[15] V. Efrati, C. Limongelli, and F. Sciarrone, "A data mining approach to the analysis of students' learning styles in an e-learning community: A case study," in *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge: 8th International Conference, UAHCI 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014, Proceedings, Part II 8*, pp. 289–300, Springer, 2014.

[16] S. Kausar, X. Huahu, I. Hussain, Z. Wenhao, and M. Zahid, "Integration of data mining clustering approach in the personalized e-learning system," *IEEE access*, vol. 6, pp. 72724–72734, 2018.

[17] S. Sivanandam, S. Deepa, S. Sivanandam, and S. Deepa, *Genetic algorithms.* Springer, 2008.

[18] G. G. Mitchell, D. O'Donoghue, D. Barnes, and M. McCarville, "Generepair-a repair operator for genetic algorithms," 2003.