MINISTRY OF EDUCATION AND RESEARCH OF REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Team 1. Codify - Empowering Coding Skills

Project report

Mentor: asist. univ., Maria Rotaru
Students: Abu Koush Islam, FAF-231
Bozadji Artiom, FAF-231
Cojocari Daniela, FAF-231
Grigoraș Janeta, FAF-231
Roenco Maxim, FAF-231
Tiganescu Anastasia, FAF-231

Chișinău, 2024

Abstract

The project named “Codify” was developed by the students Anastasia Tiganescu, Islam Abu Koush, Daniela Cojocari, Artiom Bozadji, Janeta Grigoras, Maxim Roenco from Technical University of Moldova.

This project consists of 5 chapters, Introduction, Research of the Problem, System Evaluation, Conclusion and Bibliography.

The paper introduces Codify, a web application designed to provide structured, interactive, and personalized learning experiences in programming languages. Through Codify’s user-friendly interface and innovative features, such as structured learning pathways, interactive assessments, and immersive coding environments, learners can acquire programming skills efficiently and effectively. By prioritizing engagement and adaptability, Codify aims to empower individuals to thrive in today’s digital economy, bridging the gap between the demand for programming skills and the challenges of modern learning environments.

Keywords: coding skills, technology, programming languages, education, learning experience

Content

Introduction

This is Codify, a groundbreaking solution designed to revolutionize programming education. This paper provides an in-depth exploration of Codify's key features and its potential to transform the way individuals learn programming languages.

We begin by addressing the prevalent issue of ineffective methods for teaching programming languages, especially given the increasing need for digital skills. Codify emerges as the proposed solution to this challenge, providing dynamic progress tracking, interactive assessments, and personalized learning pathways to improve student engagement and productivity in programming education.

We then go into Alex's story, a prospective web app user who is a busy aspiring programmer. Through Alex's experience, we show how Codify's interactive assessments, structured learning paths, and immersive code editor enable him to improve his skills despite his busy schedule.

The Target Group section categorizes our audience into three main groups: young students, professionals with busy schedules, and tech enthusiasts. User personas like Sarah, David, and Alex offer insightful feedback guiding Codify's feature development to cater to each group's unique needs and motivations.

A comprehensive Comparative Analysis underscores Codify's dominance in providing high-quality programming instruction. Amidst competitors like Sololearn, Mimo, and others, our web app is set apart by its personalized learning, strong feedback systems, and cutting-edge features like real-time coding duels and integrated code editing.

This report also includes UML diagrams and visual representations of our web app, providing a detailed view of Codify's functionality and user interface. The UML diagrams, including use case, activity, and sequence diagrams, offer a visual breakdown of user interactions and system processes. Furthermore, Codify's user experience is illustrated by pictures of the interface, which facilitate comprehension of the web app's functionality and how users interact with it.

Lastly, the Conclusions highlight how Codify is more than just another web app; it's an innovative tool empowering individuals to master programming languages.

Research of the Problem

Problem Description

Digital skills are now essential for success in a variety of career paths in our more technologically advanced and modern world. Given the prevalence of computers and software in almost every industry, coding stands out as one of these skills that is especially valuable. As such, there is a constant need for people who are multilingual in programming languages. However, engagement and efficiency remain a major challenge in the field of programming education, even with the abundance of online learning resources. Few students actually stick to their learning objectives and advance in their programming language learning. But why should we pay attention to this widespread problem?

First and foremost, the ability to code is now regarded as a crucial talent and can be a good predictor of professional success. According to a study focused on finding and classifying different web approaches and methods of promoting and learning coding skills called "Coding skills as a success factor for a society" (Pauliina Tuomi, Jari Multisilta, Petri Saarikoski & Jaakko Suominen) [1], one of the most promising strategies to boost productivity in the public sector is digitalization, which can transform the economy by creating new jobs related to innovation. It is highlighted that having programming skills is crucial for the advancement of our society. Ignorance of the importance of being proficient in coding not only prevents people from pursuing profitable careers but also impedes personal development. This problem is made worse by a lack of resources and assistance, which makes it more difficult for people to adjust to the changing needs of the digital age.

Secondly, as our world rapidly advances in technology, people are having to deal with busier schedules and shorter attention spans. The average attention span has decreased from 12 seconds in 2000 to 8.25 seconds in 2015, as shown by Statistic Brain [2]. It is still imperative to learn continuously despite these time constraints and cognitive challenges. Nevertheless, conventional learning resources frequently become ineffective due to their inability to adjust to the changing needs and preferences of today's learners. For educational materials to be effective and relevant in a time when information overload is the norm, engagement, and efficiency must be given top priority. This emphasizes how vital it is to give learners effective and engaging learning materials that are suited to their contemporary lifestyles and cognitive capacities. Failing to do so runs the risk of making traditional learning web approaches outdated in the face of quickly changing digital environments.

Figure 1 - Average attention spans

To sum up, traditional teaching methods usually fail to effectively engage and empower learners., even though programming skills are in high demand in today's digitally connected world. As mentioned previously, being able to code is not only a practical ability but also an essential tool in the contemporary economy, crucial for both career growth and creativity. But the persistent issue is that there aren't many interesting and useful ways to teach programming languages, especially to modern, hectic individuals. The development of efficient and captivating learning platforms must be the top priority to satisfy the demands of today's learners and solve this urgent problem. By doing this, we can make sure that people have the resources they need and they can open doors to new opportunities for both professional and personal development.

Problem Analysis

Amidst the growing necessity for digital skills in today's world, the demand for learning programming languages is skyrocketing across various sectors. However, traditional teaching methods for programming often fall short of engaging modern learners and ensuring effective knowledge retention. Through a thorough analysis, it has been concluded that the stated problem belongs to the domains of Education and Technology/Programming Instruction.

To begin with, ineffective teaching methods can significantly impact students' ability to learn and engage with course material. Novice programmers, in particular, are likely to find programming challenging due to its perceived complexity and the amount of information they must retain at an early stage of their learning. According to recent research conducted by Kadar et al. (2021) [3], one of the primary difficulties students face when learning to program is syntax. This difficulty can lead to students taking shortcuts, such as plagiarism or relying on sympathy from instructors, which ultimately undermines their education and

future career prospects. Therefore, it is crucial for educators to adopt effective teaching strategies that help students overcome these challenges and foster a deep understanding of programming.

In addition, ineffective teaching methods may also result in low retention rates and poor academic performance among students. For example, a study conducted by Rosmina et. al.(2012) [4], showed that one of the factors that lead to poor performance in programming is "Teaching methodology is less effective". This highlights the importance of using effective teaching methods that can better engage and motivate students to learn, leading to better academic outcomes. Furthermore, it is worth noting that if this problem persists, it can cause students to lose interest in the subject and ultimately drop out of the class. This is a serious problem as shown by The Edmund de Rothschild Foundation, led by Dr. R. Eitan Regev [5], which in a recently published study revealed that the dropout for STEM fields, especially computer science is web approximately 30% (10% higher than the general average).

Moreover, in today's technology sector, busy schedules present significant challenges for programming education. According to Kaseya's [6] report, 62% of the IT professionals interviewed reported having to work holidays and weekends, 46% reported pulling all-nighters, and 39% said they had to work 50+ hour weeks. Additionally, Golden Steps [7] mentions that the average attention span of a human has decreased from 12 seconds to 8.25 seconds in the last two decades. These statistics highlight the urgent need for innovative web approaches in programming education to accommodate modern learners' busy lifestyles and limited attention spans

Furthermore, this problem has a significant impact on the Technology domain. Firstly, ineffective teaching methods may hinder the development of essential programming skills, such as problem-solving and algorithmic thinking, which are crucial for success in the tech industry. According to a report done by the Royal Society [8], over 60% of new jobs in STEM fields require computational thinking skills, which underscores its importance in today's digital world.

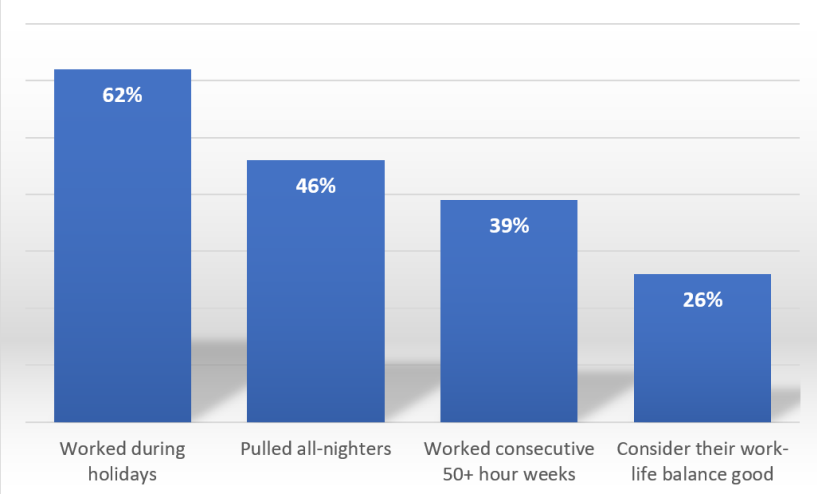

Figure 2 - Report regarding jobs in STEM

Moreover, ineffective teaching of coding can lead to a lack of innovation in the technology sector. According to the U.S. Bureau of Labor Statistics [9], employment in computer and information technology occupations is projected to grow 11% from 2019 to 2029, much faster than the average for all occupations. This highlights the need for qualified specialists to develop creative solutions to complex problems and keep up with the pace of technological advancement.

As a further matter, ineffective methods of teaching might suppress students to limited career opportunities. A study conducted by the Association for Computing Machinery (ACM) [10] found that students' confidence in their coding abilities significantly influences their pursuit of careers in technology-related fields. In the study conducted by Sarpong et al. (2013) [11], 77% of students found command style teaching often used in teaching programming affects performance and should not be used. Ineffective teaching methods that fail to instill confidence and competence in coding skills can dampen students' enthusiasm for pursuing careers in the technology sector, perpetuating a cycle of limited opportunities and underrepresentation in the industry.

In conclusion, the challenges posed by ineffective teaching methods and busy schedules in programming education underscore the urgent need for innovative web approaches within both the Education and Technology domains. As highlighted by research findings and statistics, traditional teaching methods often fail to engage modern learners and accommodate their busy lifestyles, leading to low retention rates and hindering the development of essential programming skills. Therefore, it is crucial to develop innovative web approaches that cater to the specific needs of modern learners to ensure that students are equipped with the necessary skills to succeed in the modern world of programming.

Problem Statement

This analysis led to the following problem statement: the lack of engaging and effective methods to teach programming languages to busy, modern individuals poses a significant challenge in enabling widespread and efficient acquisition of programming skills in today's fast-paced world.

Proposed Solution

In the field of programming education, learners frequently encounter difficulties in navigating the complexities of coding languages and concepts, especially amidst their busy schedules. Many struggle to find comprehensive and engaging resources that cater to their learning. Recognizing this prevalent issue, Codify emerges as a revolutionary educational platform designed to tackle these challenges and empower users to master programming with ease and proficiency.

1. Structured Learning Pathways:

   Codify offers well designed learning routes that take users from the fundamentals to more complex subjects. These courses are made to be simple to follow, so learning programming concepts will progress at a steady pace. Every pathway is logically structured, building on prior information to provide a strong foundation in coding.

2. Interactive Assessments and Challenges:

   To evaluate users' comprehension and abilities, our platform offers a range of tests and coding challenges. These tests are intended to be entertaining as well as educational. Instant feedback is given to users so they can learn from their mistakes and advance their coding skills.

3. Dynamic Progress Monitoring:

   With Codify's dynamic monitoring tools, users can monitor their progress and maintain motivation. Along with their accomplishments and benchmarks, users can effortlessly view the lessons, tests, and challenges they have finished. This feature motivates users to keep learning and helps them stay on course.

4. Immersive Code Editor Environment:

   Additionally, Codify's integrated code editor allows you to get hands-on experience with coding. Code can be written, executed, and debugged right within the web app with our editor. With tools like real-time error feedback and syntax highlighting, users can concentrate on learning without being distracted.

5. Interactive Language Exploration:

   With Codify's interactive tours, users can learn about programming languages in an engaging way. These tours offer a guided exploration of the characteristics, syntax, and best practices of each language. Users can get a deeper understanding of coding languages and learn at their own pace.

6. Language-Specific Challenges and Solutions Repository:

   Furthermore, you can test your coding abilities with the many language-specific challenges offered by Codify. Users can learn through doing by having access to thorough explanations and solutions for every challenge. This feature aids in strengthening knowledge and developing problem-solving abilities.

7. Personalized Learning Scheduler:

   Codify offers a personalized learning scheduler that enables users to customize their educational journey. It is simple for them to fit coding practice into their hectic schedules because they can arrange study sessions according to their preferences and availability. All users can be assured that learning is still manageable and accessible thanks to this flexibility.

8. Comprehensive Reference Library:

Lastly, Codify's comprehensive reference library gives you access to a multitude of resources. Our collection offers helpful assistance for students at every level, ranging from brief explanations of important concepts to code snippets. Users can easily locate the data they need to overcome obstacles and reinforce their learning.

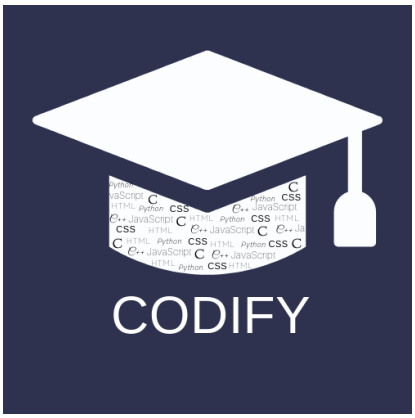

Figure 3 - Codify Logo

User story

The functions that were listed earlier serve as a focal point for overseeing the learning of programming languages. To illustrate, an example of how users can interact with them will be provided next.

Meet Alex, a budding programmer with a hunger for knowledge and a bustling schedule to balance. Seeking to master programming concepts and enhance their skills, Alex turns to Codify. Alex embarks on his learning journey by exploring Codify's meticulously designed learning routes. Starting from the basics and progressing to more advanced topics, they web appreciate the logical flow of the courses, which provide a sturdy foundation in coding while ensuring steady progress.

Eager to gauge his comprehension and abilities, Alex eagerly dives into the platform's interactive assessments and challenges. These engaging yet educational tests offer instant feedback, allowing him to learn from mistakes and refine their coding skills with each attempt. As Alex advances, he relies on Codify's dynamic progress monitoring tools to track his growth. With easy access to accomplishments and benchmarks, he stays motivated and focused on his learning goals, effortlessly navigating through completed lessons, tests, and challenges.

When it comes to practical coding experience, Alex relies on Codify's immersive code editor environment. Here, he can write, execute, and debug code directly within the web app, benefiting from real-time error feedback and syntax highlighting that helps him focus on learning without distractions. To deepen his understanding of programming languages, Alex immerses himself in Codify's interactive language exploration tours. These guided explorations offer insights into language characteristics, syntax, and best practices, allowing him to learn at their own pace and develop a deeper understanding of coding

languages.

When faced with language-specific challenges, Alex embraces the opportunity to test his skills. With access to detailed explanations and solutions for each challenge, he strengthens his knowledge and hone his problem-solving abilities, gaining confidence in his coding prowess. Despite a busy schedule, Alex effortlessly manages his learning journey with Codify's personalized learning scheduler. Customizing study sessions to fit his preferences and availability ensures that learning remains manageable and accessible, even amidst his hectic lifestyle.

Finally, Alex takes advantage of Codify's comprehensive reference library, where he finds a wealth of resources to aid his learning journey. From concise explanations of key concepts to useful code snippets, he easily locates the information needed to overcome challenges and reinforce his learning.

Through his creative utilization of Codify, Alex embarks on a fulfilling journey of learning and development, empowered to navigate the complexities of programming with confidence and skill.

Target Group

Understanding the target audience is crucial for the success of any product or service, especially in the realm of educational technology. In the context of educational platforms like Codify, defining target groups helps in designing features and content that align with the diverse learning styles and objectives of users. In the subsequent paragraphs, the three main types of users that would benefit from Codify will be presented.

The first group Codify addresses is young students and learners. These individuals prioritize flexibility and convenience in their learning experience, as they often have to balance their academic pursuits with social life and other commitments. They are interested in finding interactive and engaging methods to nurture their curiosity and creativity while improving their coding skills. Driven by their desire for personal growth and academic achievement, they are eager to explore learning platforms such as Codify to support their journey of learning.

The second group are professionals with busy schedules. This group consists of individuals who face time constraints due to their demanding careers. But they all share a common trait: they are all highly motivated by their passion for innovation. Despite their busy schedules, they are eager to acquire programming skills to enhance their professional growth. They value efficient and flexible learning solutions that allow them to integrate coding practice into their hectic lives seamlessly. Seeking opportunities for career advancement and personal development, they turn to platforms like Codify for comprehensive resources and support. With Codify, these individuals can pursue their passion for innovation with confidence and without sacrificing their professional or personal obligations.

Finally, the third group Codify addresses to are tech enthusiasts and self-learners. This group consists of individuals who are highly motivated to learn programming independently. They web appreciate the convenience of being able to learn on the go and value platforms that provide structured learning pathways and interactive assessments. Their curiosity and interest in technology drive them to seek continuous skill development and hands-on learning experiences. They prioritize personal growth and skill enhancement and utilize platforms like Codify to explore new technologies and advance their coding proficiency.

Codify's web approach to education is centered around understanding the unique needs and preferences of users. With a commitment to lifelong learning and skill development, it is a valuable resource for anyone looking to stay competitive and adapt to the changing demands of the digital age.

User Personas

The next step is to create the user personas based on the target group defined earlier. In our strategy for developing user personas, our aim is to design an web app that effortlessly caters to a wide range of programming needs, ultimately enriching the overall learning experience. Our web application serves as a personalized mentor for different user profiles navigating the world of coding. By conducting a thorough examination of individual lifestyles, coding proficiency levels, and motivations, we've meticulously crafted three distinct user personas. Each persona comes with its own set of unique characteristics and programming goals, ensuring that our web app offers a comprehensive solution for learners of all backgrounds.

These carefully crafted user personas, representing a diverse range of individuals, offer a vivid snapshot of the unique needs, aspirations, and challenges that the possible users face. Through Codify, each persona will gain access to structured learning pathways, interactive assessments, immersive code editor environments, and personalized learning schedules tailored to their busy lifestyles.

User Persona 1: Sarah

Meet Sarah Grace, a 21-year-old proficient multitasker, seamlessly juggles her studies, social life, and part-time job with the help of her smartphone. Codify, an innovative learning web app, becomes her go-to companion, allowing her to access structured learning pathways, interactive assessments, and dynamic progress monitoring anytime, anywhere. With its immersive code editor and language exploration features, Sarah can enhance her programming skills on the go, even during her busiest days. The web app's personalized learning scheduler ensures that she can fit coding practice into her hectic schedule, while the comprehensive reference library provides valuable resources for overcoming obstacles and reinforcing her learning. Thanks to Codify, Sarah stays ahead in her academic pursuits while staying connected and organized in her fast-paced lifestyle.

Figure 4 - User persona 1

User Persona 2: David

Introducing David Simpson, a 32-year-old marketing manager driven by a relentless passion for creativity and innovation. With Codify, David gains access to structured learning pathways, interactive assessments, and immersive code editor environments tailored to his busy lifestyle.

Figure 5 - User persona 2

Codify's personalized learning scheduler enables David to seamlessly integrate coding practice into his schedule, while its comprehensive reference library provides invaluable resources for fueling his creativity and driving his career forward. With Codify by his side, David embarks on a transformative journey of self-discovery and innovation, equipped with the tools and support to realize his full potential.

User Persona 3: Alex

With Codify, Alex Dunphy, a 16-year-old explorer with a penchant for collecting fallen leaves and dreaming of distant galaxies, finds a haven for her boundless curiosity in the quiet suburbs. From her cozy home, Alex delves into structured learning pathways, interactive assessments, and immersive code editor environments, nurturing her love for exploration and creativity while honing her coding skills. With Codify as her guide, Alex's adventures take on new dimensions as she discovers the wonders of coding and technology, transforming her dreams into reality one line of code at a time.

Figure 6 - User persona 3

System evaluation

Comparative Analysis

Nowadays, programming has become one of the most valuable professions, with countless individuals eager to dive into the realm of programming. This amount of interest highlights the need for an web app that can guide beginners from ground zero or help seasoned coders refine their skills. Our web app steps into this arena, distinguishing itself from rivals like Sololearn, Mimo, Enki, Encode, and Programming Hub. In the next paragraphs, we'll explore what makes our web app unique and why it's the best choice.

While competitors like Sololearn, Mimo, Enki, Encode, and Programming Hub offer structured learning paths, they often lack the personalized touch that sets our web app apart. Unlike these platforms, which provide predefined courses without much flexibility, our web app empowers users to customize their learning journey based on their individual skill levels and aspirations. This personalized web approach ensures maximum engagement and efficacy, allowing users to chart their unique path from novice to adept coder.

When considering quizzes and assessments, it's worth noting that while Sololearn, Encode, and Programming Hub do incorporate these features, the feedback they offer tends to be somewhat limited. Mimo and Enki, on the other hand, provide feedback that may vary in depth and effectiveness. In contrast, our web app takes a more proactive web approach, ensuring that quizzes and assessments are not just integrated seamlessly into the learning experience but also accompanied by thorough feedback and guidance. This ensures that users have the tools they need to track their progress effectively and make meaningful improvements to their skills.

Our web app prioritizes user engagement and learning progression through robust progress tracking features, enabling users to monitor completed lessons, quizzes, and coding challenges. This comprehensive tracking not only motivates users to continue learning but also provides valuable insights into their skill development journey. In contrast, competitors like Sololearn offer basic progress tracking, lacking detailed analytics and personalized feedback. Mimo and Enki incorporate progress tracking functionality, but the effectiveness of feedback may vary, potentially limiting users' ability to gauge their progress accurately. Encode integrates progress tracking features, but users may experience inconsistencies in feedback. Similarly, Programming Hub includes progress tracking functionalities, with potential variations in feedback effectiveness.

Our web app offers a unique advantage with its integrated code editor, providing learners with a seamless platform to write, execute, and debug code directly within the web app. This interactive tool features syntax highlighting, auto-completion, and real-time error feedback, enhancing the learning experience

and allowing users to practice coding in a supportive environment. Unlike competitors, who may rely on external resources for coding practice, our web app empowers users with a comprehensive solution that integrates all essential coding functionalities within a single platform.

Both our web app and Programming Hub provide interactive language tours, but our web app takes it a step further by offering individualized tours for each programming language. These interactive tours serve as guided learning experiences, taking learners through essential language features, syntax rules, and common programming patterns in a user-friendly manner.

Furthermore, our web app offers a comprehensive calendar system for scheduling tasks and customizing study intensity. While our competitors provide basic scheduling features, our web app stands out with customizable study intensity options and incentives for consistent usage. Sololearn tracks progress but lacks personalized intensity settings and incentives. Similarly, Mimo and Enki offer scheduling but may lack customization and motivation features. Programming Hub includes scheduling but may not offer as robust a system as our web app.

Our web app offers integrated reference books, presenting pieces of code similarly to documentation or programming books directly within the web application. Similarly, Mimo also provides reference books, offering users access to code snippets and programming resources within the web app. However, our web app distinguishes itself by seamlessly integrating these resources into the learning experience, ensuring users have instant access to valuable coding knowledge while engaging with course materials.

Our web app introduces an innovative feature where users can engage in coding duels, akin to a game environment. In these duels, users challenge each other to real-time coding challenges within a set time limit, earning points based on accuracy and speed. Similarly, Sololearn and Mimo offer coding challenges and exercises, but they lack a specific ”code duels” feature like ours. While users can participate in challenges and quizzes on these platforms, they are typically asynchronous and not real-time competitions against other users.

Revising all this points, our web app distinguishes itself from competitors like Sololearn, Mimo, Enki, Encode, and Programming Hub by offering a personalized learning experience, robust feedback mechanisms, and innovative features like integrated code editing and real-time coding duels. While competitors may offer structured learning paths and basic progress tracking, our web app stands out with its emphasis on customization, engagement, and interactive learning tools.

	Our App	Sololearn	Mimo	Enki	Encode	Programming Hub
Structured Learning Paths	✓	✓	✓	✓	✓	✓
Quizzes an Assessments	✓	✓	✓	✓	✓	✓
Progress Tracking	✓	✓	✓	✓	✓	✓
Interactive Code Editor	✓	✗	✗	✗	✗	✗
Interactive Language Tours	✓	✗	✗	✗	✗	✓
Schedule	✓	✓	✓	✓	✗	✓
Reference Books	✓	✗	✓	✗	✗	✗
Coding Duels	✓	✓	✓	✗	✗	✗

Figure 7 - Comparative Analysis

System Modelling

Here we present the Unified Modeling Language (UML) diagrams that detail the key features and interactions within our learning platform. These diagrams provide a visual representation of the system's functionality and user interactions. We included several types of UML diagrams: use case diagrams, activity diagrams, and sequence diagrams. The use case diagrams illustrate the primary functions and actions available to the user. The activity diagrams depict the step-by-step workflows for various features. The sequence diagrams show the interactions between the user and the system over time. You will obtain a solid understanding of how users interact with the platform, advance through classes, accumulate and utilize in-web app awards, and personalize their educational journey by studying these illustrations. A brief explanation that explains the underlying mechanisms and system behaviors is included with each diagram.

Use Case Diagram

The use case diagram for the learning platform highlights the key interactions: registration, language selection, lesson progression, quiz completion, earning and using diamonds, and adjusting settings.involves several key activities for the learner. The learning platform allows learners to register by entering their name and creating a password, ensuring future access. Once registered, learners select one of six programming

languages (Python, C, C++, CSS, JavaScript, HTML) to begin with the first lesson unlocked if they are beginners. They progress through lessons sequentially, taking quizzes at the end of each lesson. Passing quizzes unlocks the next lesson and rewards learners with diamonds, which can be used to unlock more lessons or access challenges. Learners can also access settings to change their learning language, adjust sound, and customize their experience.

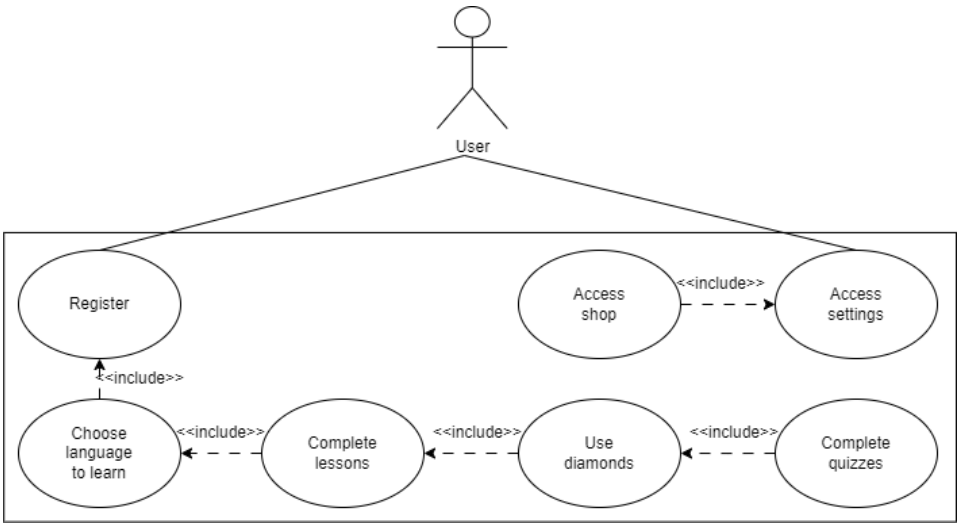

Figure 8 - Use Case diagram

Activity Diagrams

The activity diagram for the registration feature begins with the learner inputting their name and password. The system then checks if the provided credentials are valid. If the input is valid, the system stores the registration information, and the learner is successfully registered and logged in. If the input is invalid, the learner is prompted to re-enter their name and password until valid credentials are provided. This description outlines the steps: inputting name and password, validating credentials, storing registration information, and handling invalid input by prompting for re-entry.

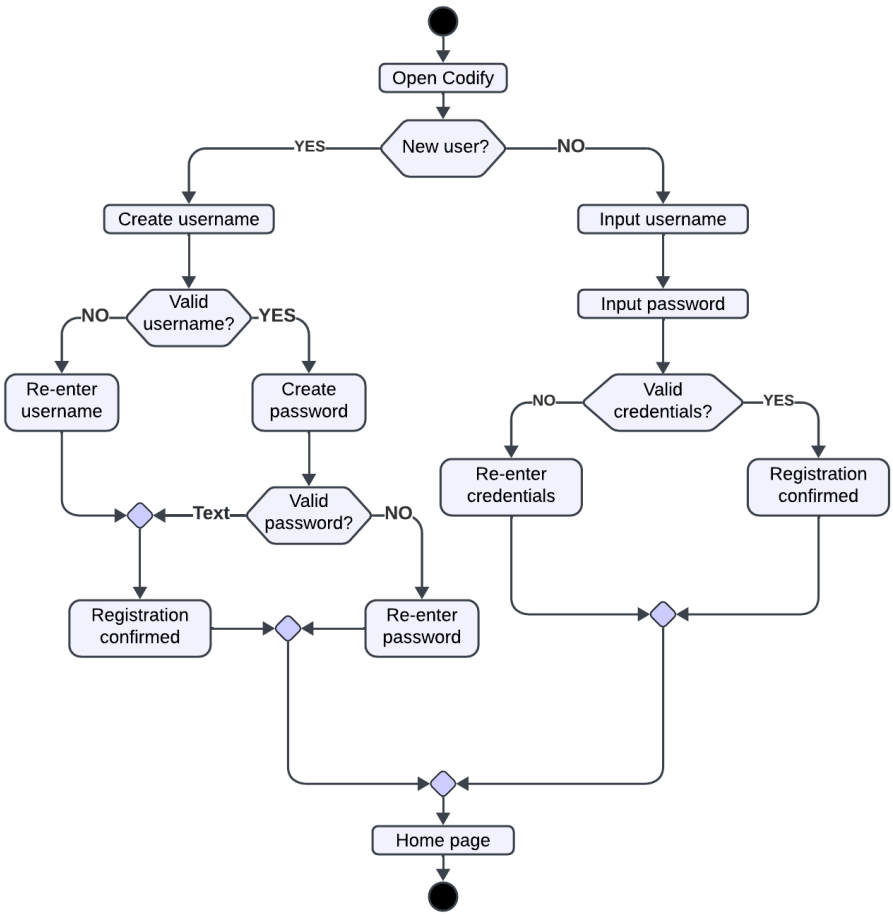

Figure 9 - Activity Diagram - Login

The activity diagram for the access account feature begins with the user logging into their account. Once logged in, the user can choose to view their statistics and success, check the number of hearts and diamonds they have, change account settings, and view their achievements. This description outlines the steps: logging into the account, viewing statistics and success, checking hearts and diamonds, changing account settings, and viewing achievements.

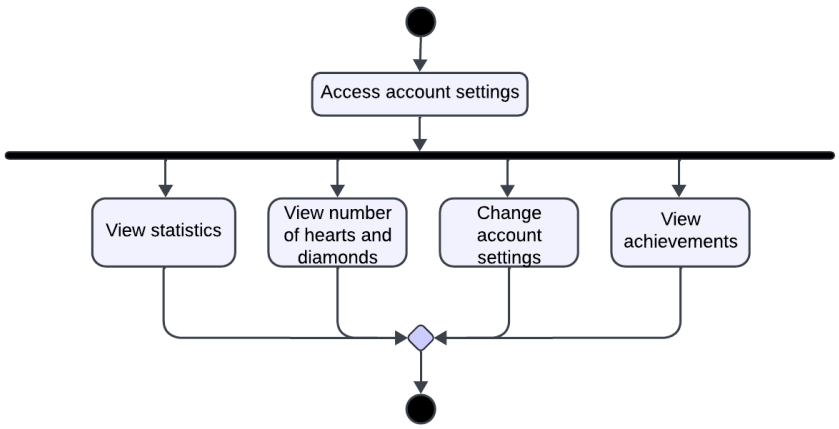

Figure 10 - Activity Diagram - Access Account

The activity diagram for the select lesson feature starts with the user choosing a programming language. The system then checks whether the user has enough hearts to access the lesson. If the user has sufficient hearts, they can proceed to the lesson. If not, the user is prompted to buy new hearts. Once the user has enough hearts, they can access the chosen lesson. This description outlines the steps: choosing a programming language, checking heart availability, prompting to buy hearts if necessary, and accessing the lesson once enough hearts are available.

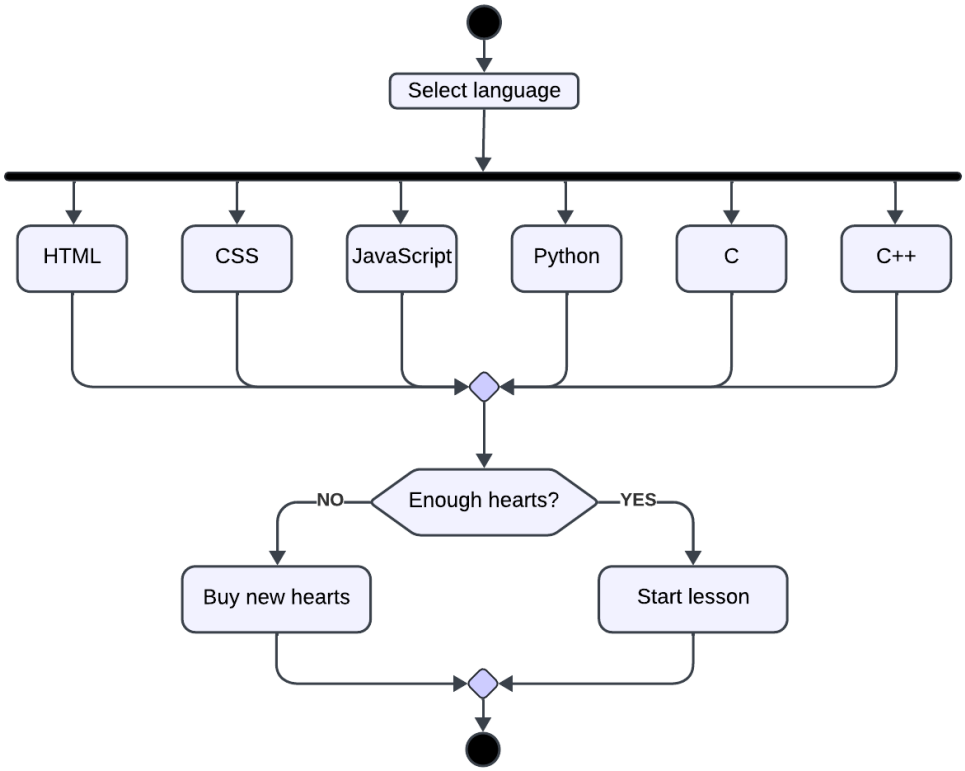

Figure 11 - Activity Diagram - Select Lesson

The activity diagram for the complete a lesson feature begins with the user choosing a programming language and then selecting the desired lesson. If the lesson is not unlocked, the user must complete the preceding lesson first. Once the lesson is unlocked, the user checks out the ”What You’ll Discover” page and then proceeds to the contents page. After finishing the lesson content, the user performs a quiz. If the quiz score is 50% or higher, the user earns diamonds and unlocks the next level. If the score is below 50%, the user must retake the quiz. This description outlines the steps: choosing a language, selecting a lesson, completing preceding lessons if necessary, reviewing the ”What You’ll Discover” page, studying

the contents, taking the quiz, and earning diamonds and unlocking the next level if the quiz is passed, or retaking the quiz if not.

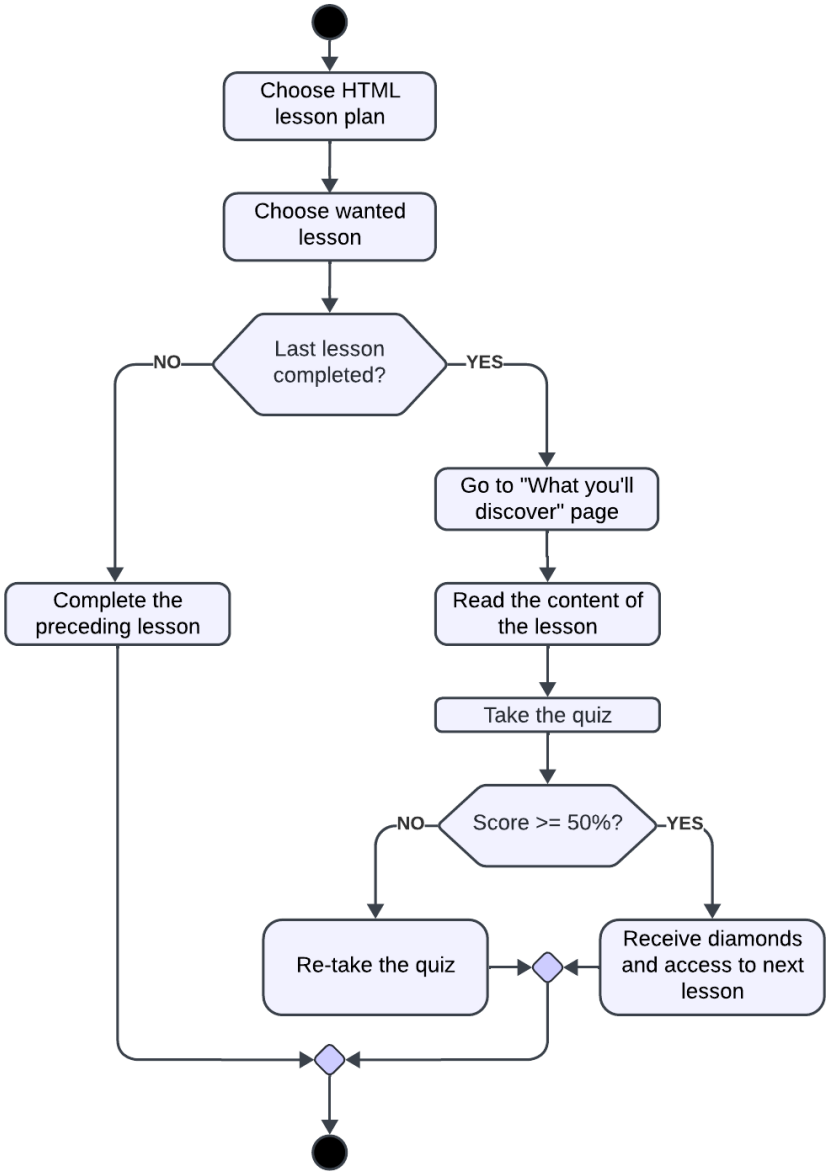

Figure 12 - Activity Diagram - Complete Lesson

The activity diagram for the access shop feature begins with the user entering the shop. Inside the shop, the user can choose to exchange diamonds for hearts, exchange diamonds for boosts, or exchange diamonds for customizations. This description outlines the steps: entering the shop, exchanging diamonds for hearts, exchanging diamonds for boosts, and exchanging diamonds for customizations.

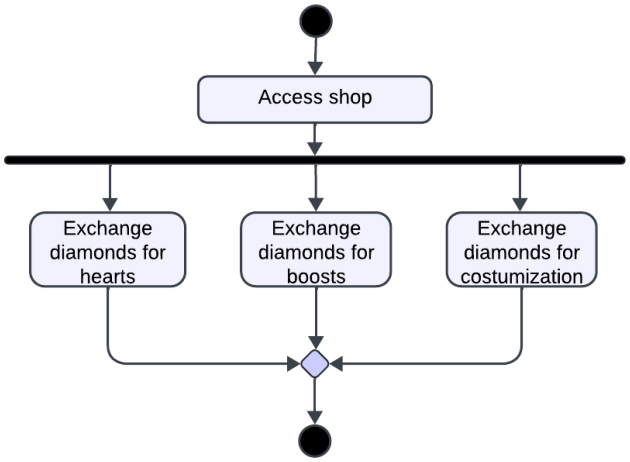

Figure 13 - Activity Diagram - Access Shop

Sequence Diagrams

The sequence diagram for the login feature begins with the user inputting their username and password into the login form and submitting it. The system then receives the credentials and checks them against stored user data. If the credentials are valid, the system authenticates the user and grants access to their account. The user can then access various features such as viewing their statistics and success, checking hearts and diamonds, changing account settings, and viewing achievements. If the credentials are invalid, the system sends an error message prompting the user to re-enter their credentials. This description captures the interaction between the user and the system during the login process, including input submission, validation, successful login, and handling invalid credentials.

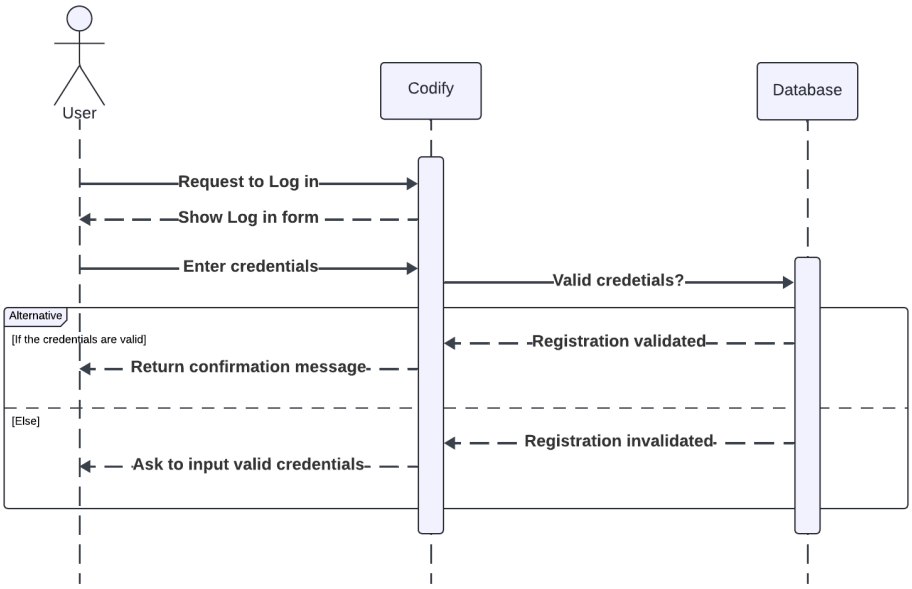

Figure 14 - Sequence Diagram - Login

The sequence diagram for selecting and completing lessons begins with the user choosing a programming language. The user then selects the desired lesson. If the lesson is not yet unlocked, the system prompts the user to complete preceding lessons first. Once the lesson is unlocked, the user accesses and reviews the ”What You’ll Discover” page, followed by the lesson content. After completing the content, the user takes a quiz. The system evaluates the quiz; if the score is 50% or higher, the user earns diamonds and the next lesson is unlocked. If the score is below 50%, the system prompts the user to retake the quiz. This description outlines the interaction between the user and the system during the processes of selecting and completing lessons, including choosing a language, selecting a lesson, completing prerequisites, reviewing content, taking a quiz, evaluating performance, and unlocking subsequent lessons or retaking the quiz as necessary.

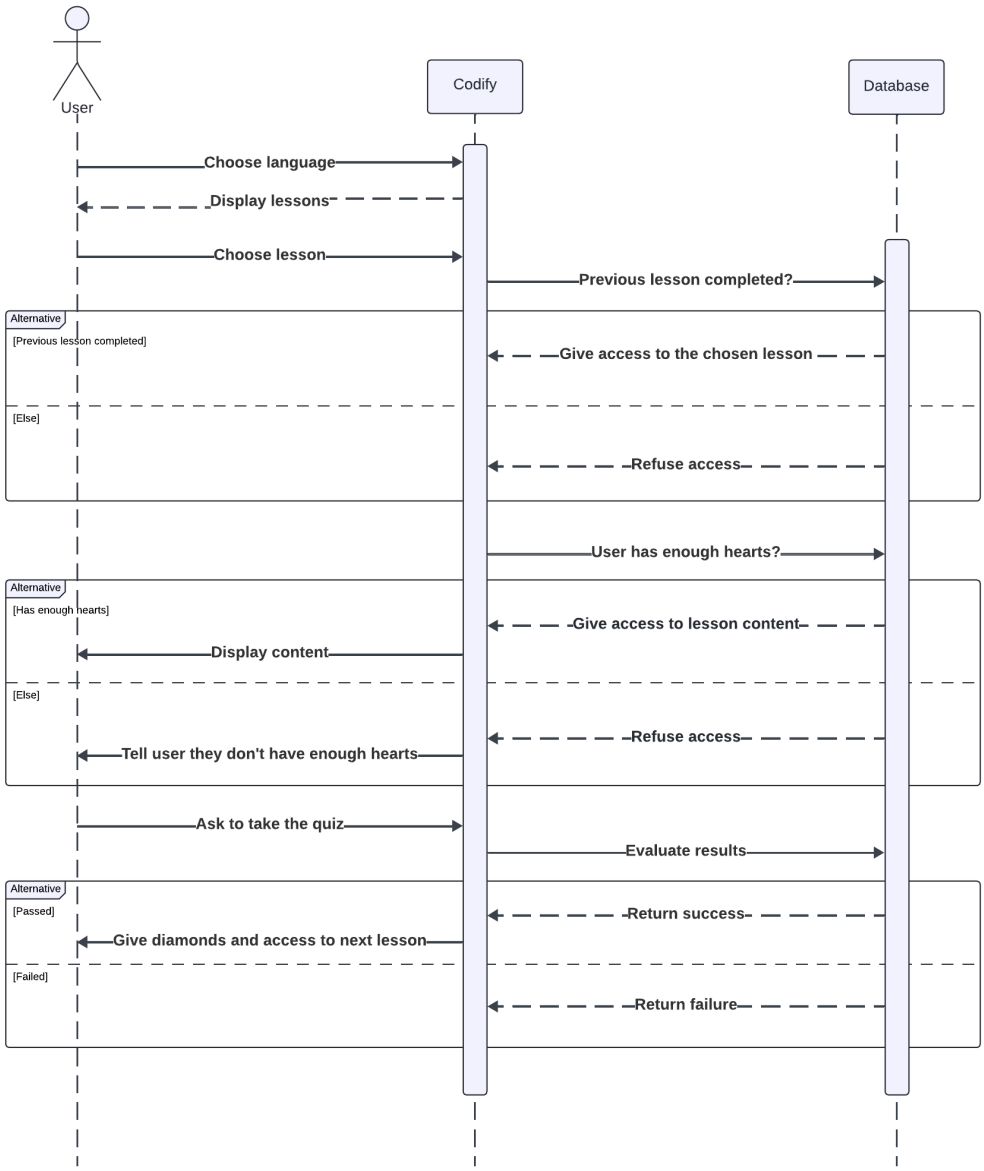

Figure 15 - Sequence Diagram - Select & Complete Lesson

The sequence diagram for the access shop feature starts with the user entering the shop. Once inside, the user can choose to exchange diamonds for hearts, boosts, or customizations. The user selects the desired exchange option, and the system checks if the user has enough diamonds. If sufficient diamonds are available, the system processes the exchange and updates the user's inventory accordingly. If there are not enough diamonds, the system prompts the user with an error message. This description outlines the interaction between the user and the system during the shop access process, including selecting an exchange option, checking diamond availability, processing the exchange, and handling insufficient diamonds.

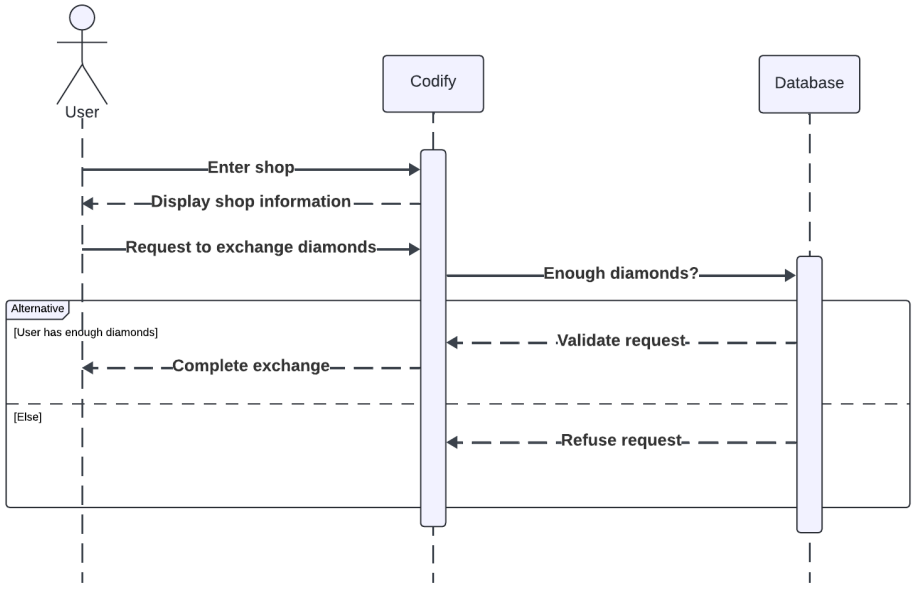

Figure 16 - Sequence Diagram - Access Shop

Solution Preview

The Solution Preview section showcases the functionality and user interface of Codify, an innovative educational platform designed to enhance programming skills. This section includes screenshots of the web application's various features, providing a visual representation of the user experience. Each screenshot is accompanied by a description that explains the purpose and functionality of the feature shown. The goal is to give a comprehensive overview of how Codify works and how it can benefit users in their journey to learn programming.

Codify is designed with a user-friendly interface and offers features such as structured learning pathways, interactive assessments, dynamic progress monitoring, interactive language exploration and language-specific challenges. These features are intended to make learning programming engaging, efficient, and adaptable to individual needs.

Through the screenshots and descriptions in this section, you will see how Codify supports users in mastering programming languages, from beginners to advanced learners. This preview aims to highlight

the web app's ease of use, and effectiveness, and how it stands out as a powerful tool for anyone looking to improve their coding skills.

The Register Page:

On the Register Page, users are prompted to enter their desired username and password to create a new account. This step is essential for accessing the web application and personalizing the learning experience. By creating an account, users can track their progress, access saved data, and continue their learning journey seamlessly across different devices.

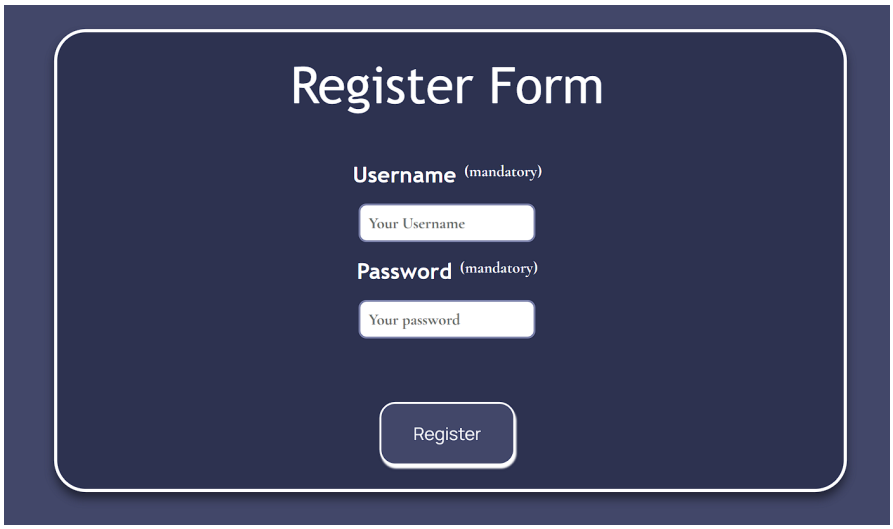

Figure 17 - The Register Page

The Home Page:

The Home Page serves as the central hub of the web application. Here, users can access various features such as settings, profile information, and learning modules. It provides a streamlined and intuitive interface where users can quickly navigate to their desired section, whether it's updating their preferences, checking their learning stats, or diving straight into the courses.

Figure 18 - The Home Page

The Settings:

Within the Settings tab, users have the flexibility to customize their web app experience. Options available include changing the language, toggling sound effects, enabling or disabling swipe mode, and logging out. This page is designed to enhance user convenience and ensure the web application meets individual preferences and accessibility needs.

Figure 19 - The Settings

The Profile:

The Profile page provides users with a detailed overview of their progress within the web application. Users can view their statistics, see the percentage of course completion, and access the shop. This page is essential for users to monitor their achievements and manage their learning goals effectively.

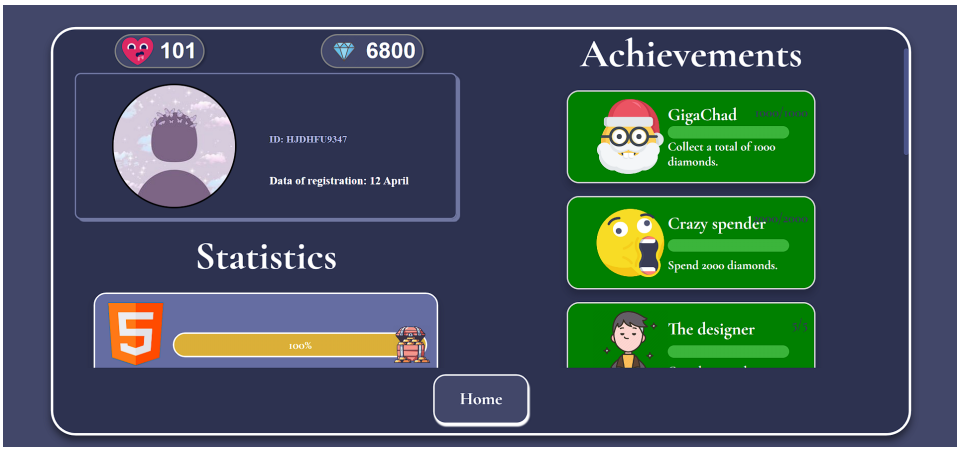

Figure 20 - The Profile

The Shop:

In the Shop section, users can purchase various in-web app items such as hearts (lives), boosts to accelerate their progress, and custom skins for their cursor. This feature not only adds a layer of personalization but also enhances the overall engagement by offering rewards and incentives.

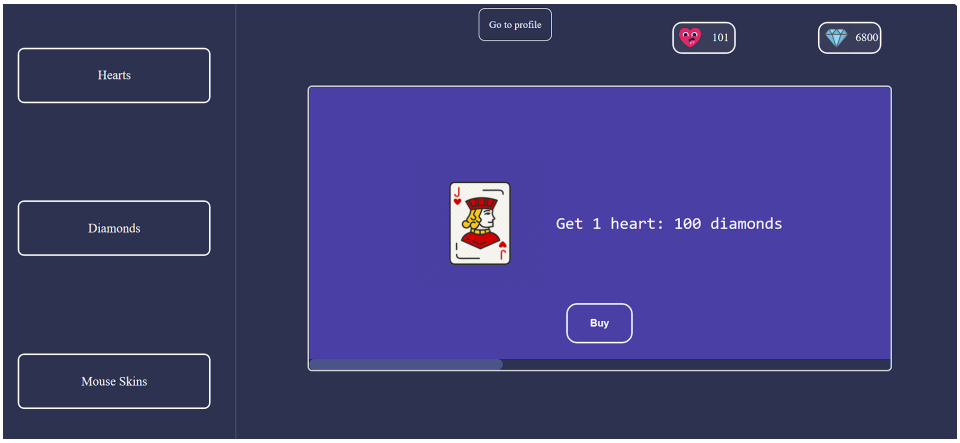

Figure 21 - The Shop

The Categories:

The Categories page allows users to select the programming course they wish to learn. There are six different options available: HTML, CSS, JavaScript (JS), C, C++, and Python. This categorization helps users to focus on specific languages and tailor their learning path according to their interests or professional requirements.

Figure 22 - The Categories

The Lessons:

Once a course is selected, users are directed to the Lessons page. Here, they can choose from a series of lessons. Initially, only the first lesson is unlocked, with subsequent lessons becoming available upon successful completion of the preceding ones. This structured web approach ensures a step-by-step learning process, building on knowledge progressively.

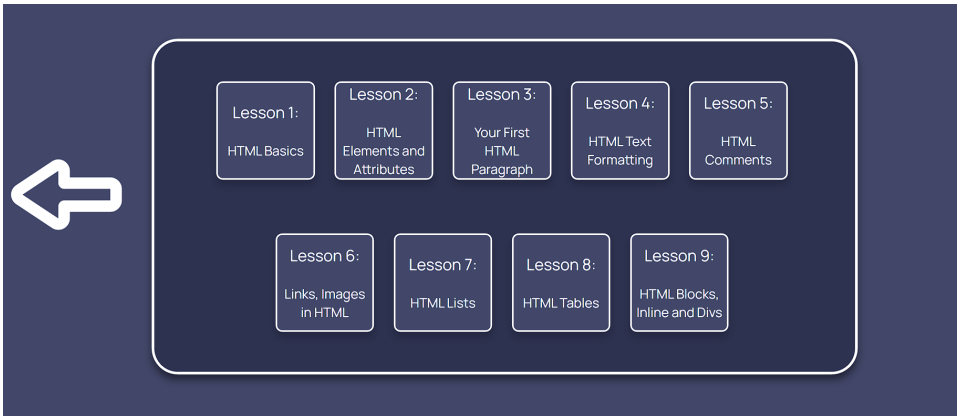

Figure 23 - The Lessons

Lesson Introduction:

The Lesson Introduction page provides users with a brief overview of the topics covered in the selected lesson. This introductory content prepares users for what they are about to learn, setting the context and objectives for the lesson ahead.

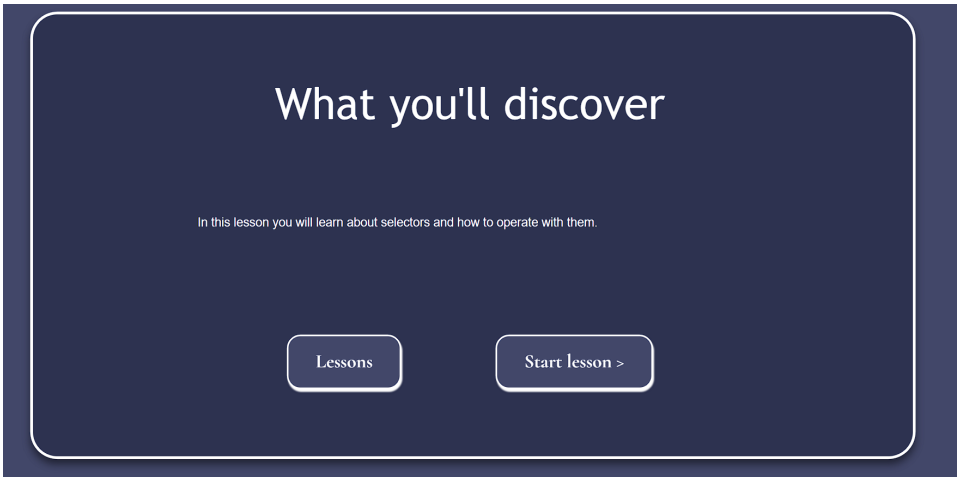

Figure 24 - Lesson Introduction

Lesson Content:

Following the introduction, users can proceed to the Lesson Content page. Here, detailed documentation related to the lesson topic is presented. This section serves as the core learning material, providing comprehensive explanations, examples, and key concepts that users need to understand the subject matter thoroughly.

Figure 25 - Lesson Content

Quiz Prompt:

After studying the lesson content, users are prompted to take a quiz. The Quiz Prompt page introduces this assessment phase, encouraging users to test their understanding and retention of the material covered in the lesson.

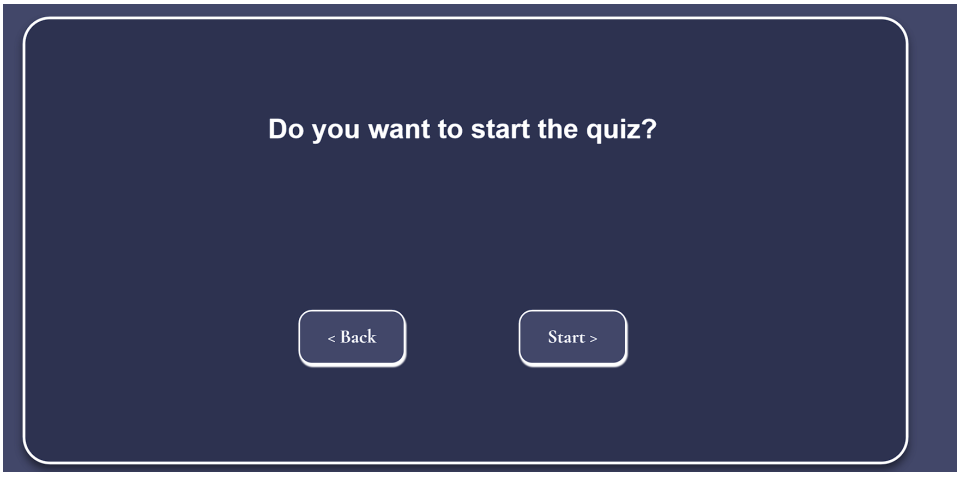

Figure 26 - Quiz Prompt

Questions:

The Questions page features multiple-choice questions designed to evaluate the user's knowledge. These questions are based on both previous knowledge and the content learned in the current lesson, ensuring a thorough assessment of the user's comprehension.

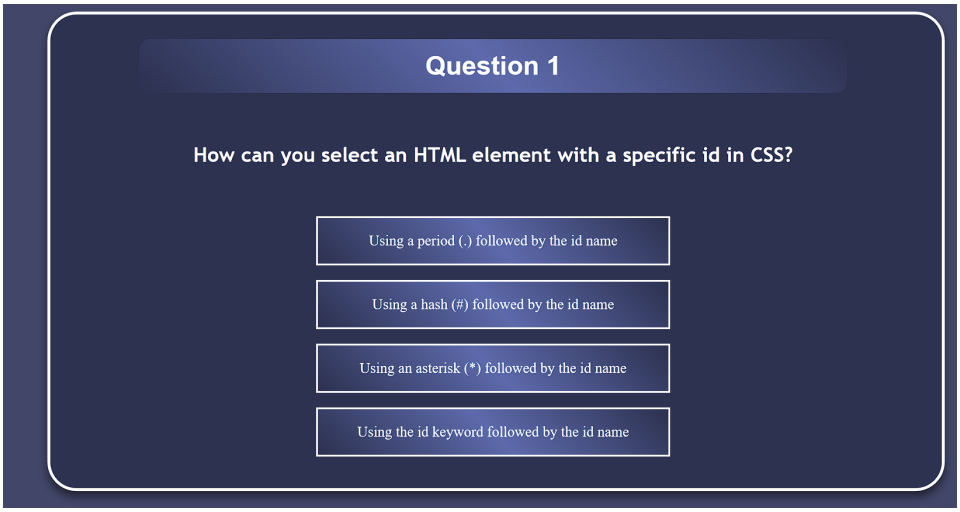

Figure 27 - Questions

The Results:

Upon completing the quiz, users are directed to the Results page. This page displays their performance, indicating whether they have answered at least 50% of the questions correctly. If they meet this threshold, they unlock the next lesson; otherwise, they must retry the quiz.

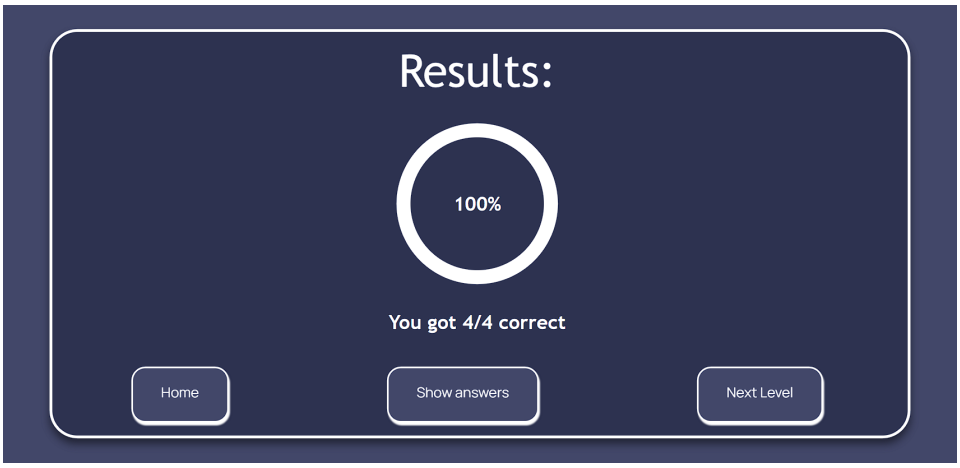

Figure 28 - The Results

The Answers:

In addition to displaying the results, the Answers page provides users with the correct answers to the quiz questions. This feedback helps users understand their mistakes and learn from them, facilitating a deeper understanding of the lesson material.

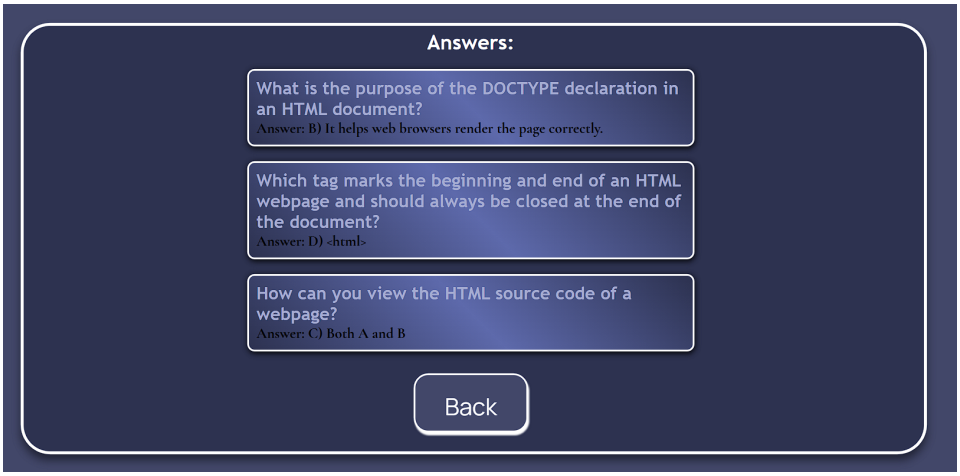

Figure 29 - The Answers

The Admin Panel:

The Admin Panel is a restricted area accessible only to administrators. In this section, admins can perform critical tasks such as adding, removing, or editing lessons, and managing other administrative functions. This panel is essential for maintaining the quality and up-to-dateness of the content provided to users.

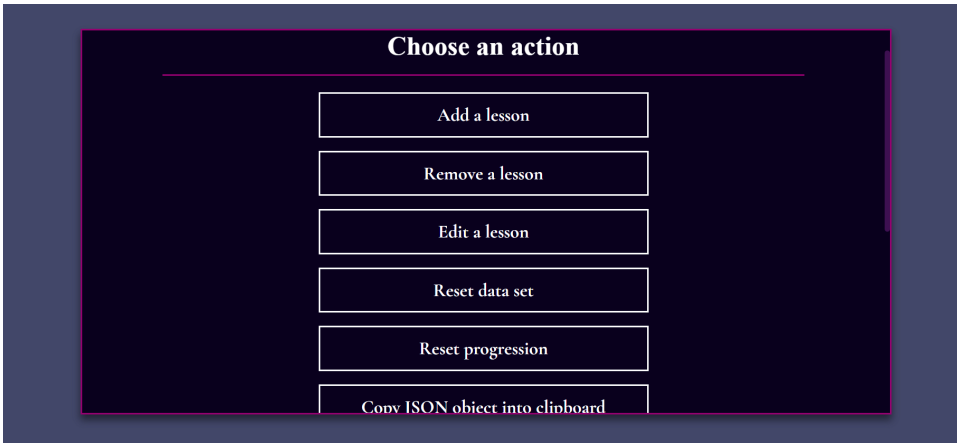

Figure 30 - The Admin Panel

Conclusions

In conclusion, our learning app is a blend between learning and entertainment, providing users with an interactive platform to learn various complex programming languages. By engaging with the app, users can better retain different structures of these languages.

One of the key strengths of our app lies in its versatility, offering a spectrum of complexity levels that accommodate users of all skill sets, from novices to seasoned developers. Whether you're taking your first steps into programming or seeking to refine your expertise, our app provides a tailored learning experience to meet your needs.

The app exclusivity relies on its features. It offers a personalized learning experience, progress tracking and coding duels. Users can embark on learning journeys with friends or family, which fosters a competitive and engaging environment. Through this web app, users can master programming skills and conveniently revisit learned concepts, ensuring continuous improvement in their programming proficiency.

In essence, our web app stands as the optimal choice for individuals of all backgrounds and expertise levels who are passionate about programming. With its dynamic and personalized web approach to learning, it promises to deliver a transformative experience tailored to your individual aspirations and objectives.

Bibliography

[1] ”Role of Mobile Learning in Education: An Overview.” Accessed February 1, 2023. https://link.springer.com/article/10.1007/s10639-017-9611-4.

[2] ”The effect of daily internet usage on a short attention span and academic performance.” Accessed February 1, 2023. https://monami.hs-mittweida.de/frontdoor/deliver/index/docId/7307/file/ANDREW+FILLMORE+BACHELOR.pdf.

[3] Kadar, Rozita & Wahab, Naemah & Othman, Jamal & Shamsuddin, Maisurah & Mahlan, Siti. (2021). A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review. International Journal of Academic Research in Progressive Education and Development. 10. 10.6007/IJARPED/v10-i3/11100. https://hrmars.com/papers_submitted/11100/a-study-of-difficulties-in-teaching-and-learning-programming-a-systematic-literature-review.pdf.

[4] Rosminah, Siti & md derus, siti rosminah & Mohamad Ali, Ahmad Zamzuri & Ali, Mohamad. (2012). Difficulties in learning programming: Views of students. 10.13140/2.1.1055.7441. https://www.researchgate.net/publication/267338258_Difficulties_in_learning_programming_Views_of_students.

[5] Edmund de Rothschild Foundation. (n.d.). Social Impact Bond for Academic Dropout Rate Reduction. Edmund de Rothschild Foundation. Retrieved from https://www.edrf.org.il/en/programs/social-impact-bond-for-academic-dropout-rate-reduction/.

[6] IT Brew. (2023, June 30). IT Professionals Report Being Overworked as Job Vacancies Hit Highs. IT Brew. Retrieved from https://www.itbrew.com/stories/2023/06/30/it-professionals-report-being-overworked-as-job-vacancies-hit-highs.

[7] Golden Steps. (n.d.). What's The Average Human Attention Span? Golden Steps. Retrieved from https://www.goldenstepsaba.com/resources/average-attention-span#:~:text=What's%20The%20Average%20Human%20Attention,span%20while%20using%20mobile%20devices.

[8] Muenks, K., DeWitt, J., McFarland, J., & Dueker, M. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Journal of Educational Psychology, 109(6), 775-785. https://www.sciencedirect.com/science/article/abs/pii/S074756321630618.

[9] U.S. Bureau of Labor Statistics. (n.d.). Computer and Information Technology Occupations. Retrieved from https://www.bls.gov/ooh/computer-and-information-technology/home.htm.

[10] ACM. (n.d.). Association for Computing Machinery. Retrieved from https://www.acm.org/

[11] Adu-Manu, Kofi & Arthur, John & Amoako, Prince. (2013). Causes of Failure of Students in Computer Programming Courses: The Teacher Learner Perspective. International Journal of Computer Applications. 77. 27-32. 10.5120/13448-1311. https://www.researchgate.net/profile/Kofi-Adu-Manu/publication/260698579_Causes_of_Failure_of_Students_in_Computer_Programming_Courses_The_Teacher_Learner_Perspective/links/5615bc3008ae983c1b422782/Causes-of-Failure-of-Students-in-Computer-Programming-Courses-The-Teacher-Learner-Perspective.pdf