**GAMIFIED LEARNING APP FOR DYSLEXIA CHILDREN**

## A PROJECT REPORT

***Submitted by***

**AARON JOEL B C (2116210701005)**

**ABESHEK SRIKANTH (2116210701009)**

**­**

***in partial fulfillment for the course***

**CS19621 – PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP**

***For the degree of***

**BACHELOR OF ENGINEERING**

***in***

## COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE**

**ANNA UNIVERSITY , CHENNAI**

**MAY 2024**

# RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

**BONAFIDE CERTIFICATE**

Certified that this Thesis titled **“GAMIFIED LEARNING APP FOR DYSLEXIA CHILDREN**” is the bonafide work of “**AARON JOEL B C (2116210701005) , ABESHEK SRIKANTH (2116210701009)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

## SIGNATURE

Dr . T.Kumaragurubaran., M.Tech.,Ph.D.,

## PROJECT COORDINATOR

AP(SG)

Department of Computer Science and Engineering , Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on **\_**

**Internal Examiner External Examine**r

# ABSTRACT

# Dyslexia, a prevalent learning disability, significantly impacts a child's ability to read, write, and spell, subsequently affecting their academic performance and self-esteem. Despite its commonality, traditional educational methods often fail to cater to the unique learning requirements of dyslexic children. These conventional approaches typically rely on text-heavy, lecture-based instruction that can be particularly challenging for dyslexic students who struggle with phonological processing, memory, and decoding skills. This project introduces a gamified learning application designed to enhance the learning capacities of dyslexic children by providing an engaging and interactive educational platform. By leveraging the principles of game design, the application creates an educational experience that is both enjoyable and effective. The gamified learning app incorporates a variety of games aimed at addressing key areas of difficulty for dyslexic learners, such as phonemic awareness, reading fluency, and spelling. Each game employs adaptive learning techniques to ensure that it meets the individual needs and learning pace of each child. Dyslexic children face unique educational challenges that can hinder their academic progress and confidence. These challenges stem from difficulties in processing language, which can affect reading comprehension, spelling, and writing skills. Traditional educational approaches often do not accommodate these needs, leading to frustration and a lack of engagement in the learning process. The gamified learning application aims to address these challenges by transforming the learning experience into a series of interactive and adaptive games. These games are specifically designed to target the areas where dyslexic children typically struggle, such as phonemic awareness, which involves the ability to hear, identify, and manipulate phonemes. By focusing on these foundational skills, the application helps children build a strong base for reading and writing.

# ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal,

**Dr. S. N. Murugesan M.E., PhD.,** and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Dr. T.Kumaragurubaran M.Tech.,Ph.D.,** for his encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

**AARON JOEL B C (2116210701005)**

**ABESHEK SRIKANTH (2116210701009)**

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | **ABSTRACT**  **LIST OF FIGURES** | **iii**  **vii** |

1. **INTRODUCTION 1**
   1. PROBLEM STATEMENT **2**
   2. SCOPE OF THE WORK **2**
   3. AIM AND OBJECTIVES OF THE PROJECT **3**
   4. EXISTING SYSTEM **3**

## LITERATURE SURVEY 5

* 1. SURVEY **5**
  2. PROPOSED SYSTEM **6**

## SYSTEM DESIGN 7

* 1. GENERAL **7**
  2. SYSTEM ARCHITECTURE DIAGRAM **7**
  3. SYSTEM FLOW DIAGRAM **8**
  4. SEQUENCE DIAGRAM **9**
  5. DEVELOPMENT ENVIRONMENT **10**
     1. HARDWARE REQUIREMENTS
     2. SOFTWARE REQUIREMENTS

## PROBLEM DESCRIPTION 11

* 1. METHODOLOGY **11**
  2. MODULE DESCRIPTION **12**

## RESULTS AND DISCUSSIONS 14

* 1. FINAL OUTPUT
  2. SOURCE CODE

## CONCLUSION AND SCOPE FOR

**FUTURE ENHANCEMENT 29**

* 1. CONCLUSION
  2. FUTURE ENHANCEMENT

## REFERENCES 31

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 3.2 | SYSTEM ARCHITECTURE | 7 |
| 3.3 | SYSTEM FLOW DIAGRAM | 7 |
| 3.4 | SEQUENCE DIAGRAM | 8 |
| 5.1 | OUTPUT | 14 |

**CHAPTER 1**

## INTRODUCTION

## Dyslexia is a neurobiological learning disorder that primarily affects an individual's ability to read, write, and spell. It is estimated that approximately 5-10% of the population is affected by dyslexia to varying degrees, making it one of the most common learning disabilities. Despite its prevalence, traditional educational approaches often fall short in addressing the unique challenges faced by dyslexic learners.

## This gap necessitates innovative educational tools and strategies tailored to the specific needs of these children. One promising approach is the integration of gamified learning applications designed specifically for dyslexic children.Dyslexia is characterized by difficulties with accurate and/or fluent word recognition and by poor spelling and decoding abilities. These difficulties typically result from a deficit in the phonological component of language, which is often unexpected in relation to other cognitive abilities and the provision of effective classroom instruction. Consequently, dyslexic children often struggle with reading comprehension, which can impede their overall academic progress and negatively impact their self-esteem.

## The impact of dyslexia extends beyond academic performance, affecting the emotional and psychological well-being of children. Many dyslexic students experience frustration, anxiety, and low self-esteem due to their struggles with traditional learning methods. This highlights the critical need for educational interventions that not only address academic challenges but also support the emotional and psychological development of dyslexic children.

## In recent years, technological advancements have provided new opportunities to support learners with special educational needs. Digital tools and applications can offer personalized learning experiences, adapt to individual progress, and provide engaging and interactive content. Gamification, the application of game-design elements in non-game contexts, has emerged as a particularly effective strategy in education. By incorporating elements such as points, badges, leaderboards, and challenges, gamified learning can increase motivation, engagement, and ultimately, educational outcomes.

## Gamified learning applications leverage the motivational aspects of games to create an engaging educational experience. These applications are designed to make learning more enjoyable and interactive, thus fostering a positive attitude towards education. For dyslexic children, gamified learning can provide an alternative approach that aligns better with their unique learning styles and needs.

## Dyslexia, a prevalent learning disability, significantly impacts a child's ability to read, write, and spell, subsequently affecting their academic performance and self-esteem. Despite its commonality, traditional educational methods often fail to cater to the unique learning requirements of dyslexic children. These conventional approaches typically rely on text-heavy, lecture-based instruction that can be particularly challenging for dyslexic students who struggle with phonological processing, memory, and decoding skills. The necessity for innovative solutions that address these specific needs is paramount in ensuring that dyslexic learners receive the support and resources they need to succeed.

## PROBLEM STATEMENT

## Dyslexia, a common yet often misunderstood learning disability, affects a child's ability to read, write, and spell, significantly impacting their academic performance and self-esteem. Traditional educational methods are predominantly text-heavy and lecture-based, which do not cater to the specific needs of dyslexic learners. These methods fail to engage dyslexic children effectively, leading to frustration, disengagement, and a lack of progress in critical literacy skills. The absence of personalized, adaptive learning tools that address the unique challenges faced by dyslexic students exacerbates the issue, highlighting the urgent need for innovative educational solutions that can provide tailored support and motivation..

## SCOPE OF THE WORK

The scope of this project encompasses the development, implementation, and evaluation of a gamified learning application specifically designed for dyslexic children. The project includes the design and creation of interactive, adaptive games that focus on improving phonemic awareness, reading fluency, and spelling skills. It also involves the integration of a reward system to motivate learners and a robust tracking mechanism to monitor progress and provide detailed analytics for educators and parents. The project aims to provide a comprehensive educational tool that not only enhances the learning experience for dyslexic children but also offers actionable insights to support tailored educational interventions. The scope extends to continuous user feedback and iterative improvements to ensure the app remains effective and engaging.

## AIM AND OBJECTIVES OF THE PROJECT

The primary aim of this project is to enhance the learning capacities of dyslexic children through a gamified educational application that makes learning engaging and effective. The specific objectives include developing interactive games that address the core challenges of dyslexia, such as phonemic awareness and reading fluency; implementing adaptive learning techniques to personalize the educational experience for each child; creating a reward system to motivate and encourage continuous learning; and establishing a comprehensive tracking system to monitor progress and provide detailed reports to educators and parents. By achieving these objectives, the project seeks to provide a supportive and stimulating learning environment that improves literacy skills and boosts the confidence of dyslexic learners.

* 1. **EXISTING SYSTEM:**

Current educational approaches for dyslexic children primarily utilize traditional methods, which include specialized instruction and tutoring focused on phonics, multisensory teaching techniques, and individualized education plans (IEPs). Specialized instruction often involves one-on-one sessions with trained educators or therapists who employ systematic, explicit teaching strategies to address the specific phonological processing difficulties faced by dyslexic learners. These methods emphasize the use of visual, auditory, and kinesthetic-tactile pathways simultaneously to enhance memory and learning of written language. Programs like Orton-Gillingham and Wilson Reading System are widely recognized for their effectiveness in supporting dyslexic students.

Despite their benefits, these traditional approaches have significant limitations. They are resource-intensive, requiring highly trained professionals and considerable time investment, which can be a barrier in under-resourced schools and communities. The individual nature of this instruction often makes it challenging to scale and implement broadly across diverse educational settings. Additionally, these methods can be perceived as monotonous and unengaging by young learners, leading to a lack of sustained motivation and interest in the learning process.

* 1. **PROPOSED SYSTEM:**

The proposed system introduces a gamified learning application designed to meet the unique needs of dyslexic children by combining educational content with game design elements. This application will feature a variety of interactive and adaptive games that target key literacy skills, such as phonemic awareness, reading fluency, and spelling. Each game will be designed to adjust to the individual learning pace and progress of each child, ensuring personalized educational support. The app will include a reward system where children can earn scores, badges, and other incentives to maintain motivation and engagement. Additionally, the system will incorporate a robust progress tracking feature that provides real-time analytics and detailed reports for educators and parents, allowing for tailored interventions and support. By integrating these elements, the proposed system aims to create an engaging, effective, and supportive learning environment that significantly enhances the educational outcomes for dyslexic children.

**CHAPTER 2**

**LITERATURE SURVEY**

## Dyslexia is a neurological condition characterized by difficulties with accurate and/or fluent word recognition and by poor spelling and decoding abilities. Shaywitz (1998) defines dyslexia as a specific learning disability that is neurobiological in origin, often resulting in poor reading fluency and reading comprehension despite normal intelligence and adequate instruction. This condition can lead to significant challenges in academic performance and self-esteem for affected children. Traditional educational practices often fail to address these unique needs, necessitating the exploration of more innovative approaches.

## Traditional interventions for dyslexia typically involve structured literacy programs. Programs such as Orton-Gillingham, Wilson Reading System, and Lindamood-Bell Learning Processes are widely recognized. Orton-Gillingham, developed in the 1930s, uses a multisensory approach to teaching reading, which has been found effective in numerous studies (Ritchey & Goeke, 2006). This method integrates visual, auditory, and kinesthetic learning pathways to enhance memory and learning of written language. Similarly, the Wilson Reading System is a structured literacy program that emphasizes phonemic awareness, decoding, and spelling through explicit, systematic instruction (Moats, 2009). Despite their effectiveness, these programs require significant resources and time, which can be a limitation in many educational settings.

## With advancements in technology, digital learning tools have been introduced to aid in the education of dyslexic children. Studies have shown that digital tools can offer individualized learning experiences and instant feedback, which are crucial for effective learning (McKnight et al., 2016). However, many of these tools are not specifically designed for dyslexic learners. For instance, general literacy apps like ABCmouse and Starfall provide engaging content but lack the specialized focus needed to address the unique challenges of dyslexia (Cheung & Slavin, 2012). These tools often do not incorporate adaptive learning technologies that can tailor the educational experience to the needs of each individual learner, making them less effective for dyslexic children.

## Gamification, the use of game design elements in non-game contexts, has gained attention as a potential solution to increase engagement and motivation in educational settings. Deterding et al. (2011) describe gamification as incorporating game mechanics, such as points, badges, leaderboards, and levels, into learning activities to make them more engaging. Research indicates that gamification can improve motivation, engagement, and educational outcomes (Hamari et al., 2014). In particular, it has been found effective in maintaining student interest and encouraging consistent practice, which is critical for dyslexic learners who often face repetitive and challenging tasks.

## There are emerging gamified learning applications designed specifically for dyslexic children. Nessy Learning, for instance, offers a suite of games and activities targeting reading and spelling skills using a structured, multisensory approach (Nessy Learning, 2017). These applications often include adaptive learning features that adjust the difficulty level based on the user’s performance, ensuring personalized learning experiences. Similarly, Touch-type Read and Spell (TTRS) combines typing with phonics to help dyslexic learners improve their reading and spelling skills. Studies show that such applications can be highly effective in improving literacy skills among dyslexic learners by providing interactive and engaging content tailored to their needs (Knight et al., 2019).

## Adaptive learning technologies play a critical role in gamified educational applications. These technologies use algorithms to analyze the learner's performance and adjust the difficulty and type of content presented accordingly (Klinkenberg et al., 2011). This approach ensures that the learning experience is personalized and responsive to the needs of each student. Research by Johnson et al. (2015) highlights the effectiveness of adaptive learning systems in improving educational outcomes, particularly for students with learning disabilities. By continuously monitoring progress and providing tailored support, adaptive learning technologies can help dyslexic students overcome specific challenges and make steady progress.

## An essential feature of effective educational tools for dyslexic children is the ability to monitor progress and provide detailed analytics. This allows educators and parents to understand the child's strengths and areas needing improvement, enabling targeted interventions. Shapiro (2011) emphasizes the importance of data-driven decision-making in special education, where continuous progress monitoring can significantly enhance the effectiveness of instructional strategies. Modern gamified learning apps often include comprehensive tracking systems that provide real-time feedback and detailed reports on the learner's progress. This data is invaluable for making informed decisions about educational interventions and support.

## The use of rewards and incentives is a common gamification strategy to enhance motivation and engagement. Research shows that providing extrinsic rewards, such as points, badges, and certificates, can significantly increase student engagement and motivation (Deci et al., 2001). However, it is essential to balance extrinsic and intrinsic motivation to ensure sustained engagement. Studies by Ryan and Deci (2000) indicate that while extrinsic rewards can boost initial motivation, fostering intrinsic motivation—where students engage in an activity for its inherent satisfaction—is crucial for long-term engagement. Effective gamified learning applications for dyslexic children should therefore include a well-balanced reward system that encourages both extrinsic and intrinsic motivation.

## Despite the promising potential of gamified learning applications, several challenges remain. Ensuring accessibility and inclusivity is critical, as not all dyslexic learners have the same needs and abilities. Customizing the content and design to cater to a wide range of dyslexic profiles is essential. Additionally, the effectiveness of these applications depends on their proper integration into the existing educational framework. Teachers and parents need to be adequately trained to use these tools effectively (Stanberry, 2016).

## Future research should focus on longitudinal studies to assess the long-term impact of gamified learning applications on dyslexic learners. Exploring the integration of advanced technologies, such as artificial intelligence and machine learning, can further enhance the adaptiveness and effectiveness of these tools. Moreover, developing more comprehensive and inclusive educational strategies that combine traditional methods with innovative digital solutions can provide a holistic approach to supporting dyslexic children.

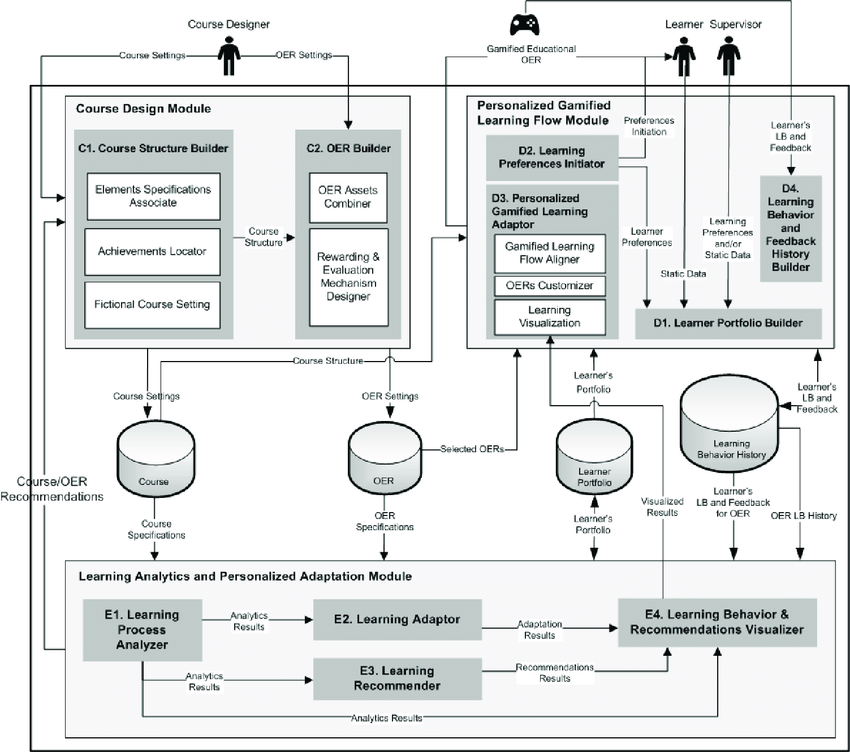
**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 GENERAL**

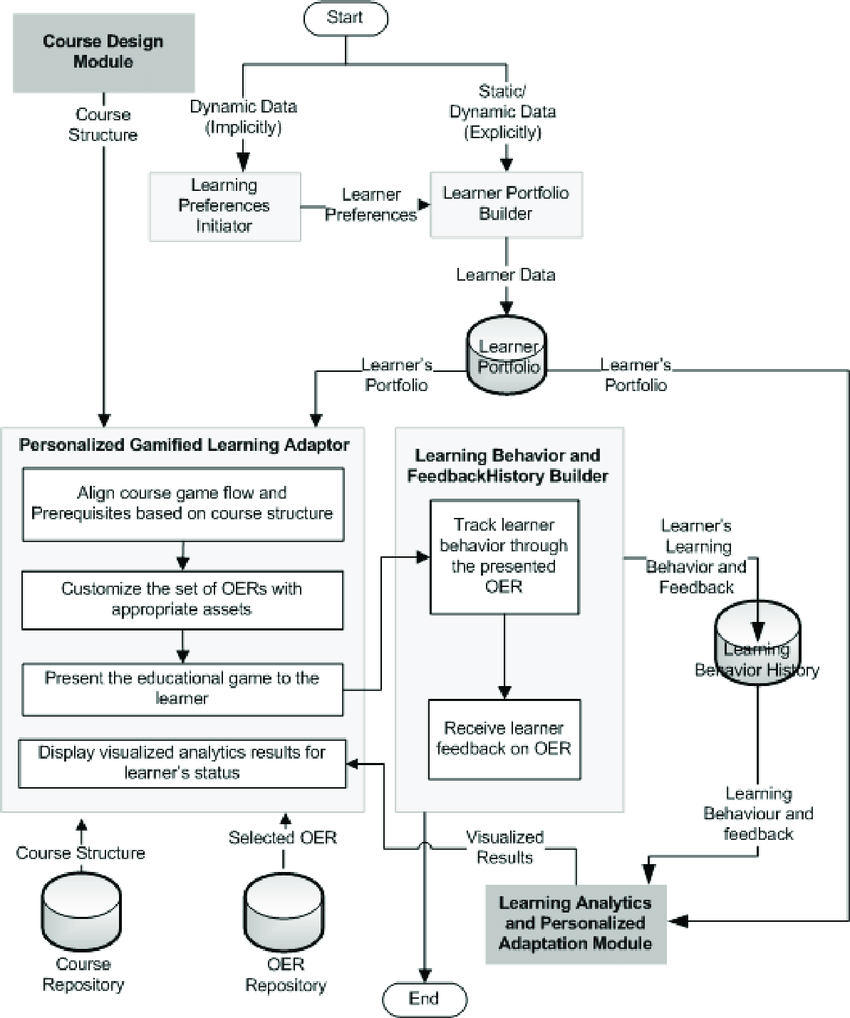
In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

## SYSTEM ARCHITECTURE DIAGRAM



**Fig 3.1: System Architecture**

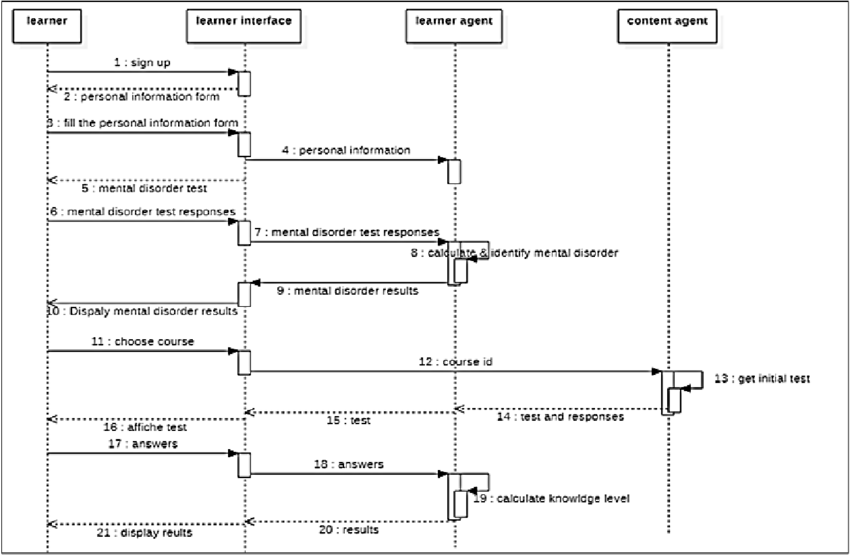
**3.2 SYSTEM FLOW DIAGRAM:**

****

**Fig 3.2: System Flow**

**3.3 SEQUENCE DIAGRAM**

A sequence diagram is a type of interaction diagram in the Unified Modelling Language (UML) that illustrates the interactions between objects or components within a system in a chronological order. It provides a dynamic view of the system's behaviour by depicting the sequence of messages exchanged between different entities over time.

****

**FIG. 3.3 Sequence Diagram**

## DEVELOPMENTAL ENVIRONMENT

* + 1. **HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the system’s implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

## Table 3.1 Hardware Requirements

|  |  |
| --- | --- |
| **COMPONENTS** | **SPECIFICATION** |
| PROCESSOR | Intel Core i5 |
| RAM | 8 GB RAM |
| HARD DISK | 512 GB |
| PROCESSOR SPEED | MINIMUM 1.1 GHz |

* + 1. **SOFTWARE REQUIREMENTS**

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team’s progress throughout the development activity.

**Python IDLE,** and **chrome** would all be required.

**CHAPTER 4**

**PROJECT DESCRIPTION**

## METHODOLODGY

## 

The project aims to develop a gamified learning application specifically designed to improve the literacy skills of children with dyslexia. Dyslexia affects a significant percentage of the population, leading to difficulties in reading, writing, and spelling, which can hinder academic progress and self-esteem. Traditional teaching methods often fail to address the unique needs of dyslexic learners, necessitating innovative educational solutions. This project leverages the principles of game design to create an interactive and adaptive learning environment that is both engaging and effective. The application features a variety of games targeting key areas of difficulty for dyslexic children, such as phonemic awareness, reading fluency, and spelling. These games are designed to be adaptive, adjusting the difficulty level based on the individual learner’s progress, ensuring a personalized learning experience. To motivate and sustain engagement, the app employs a reward system where children can earn scores, badges, and other incentives upon completing learning milestones. This gamification strategy not only makes learning fun but also encourages continuous effort and progress. Additionally, the application includes a comprehensive tracking system that monitors the child's progress in real-time. This feature allows educators and parents to access detailed reports and analytics, providing insights into the child’s strengths and areas needing improvement. These data-driven insights enable tailored educational interventions, enhancing the overall effectiveness of the learning process. The project also incorporates user feedback for continuous improvement, ensuring that the app remains responsive to the needs of its users.

## MODULE DESCRIPTION

## Interactive Learning Games:

## Description: This module comprises a series of educational games designed to target key literacy skills such as phonemic awareness, reading fluency, and spelling.

## Features: Adaptive difficulty levels, engaging graphics, and interactive elements to maintain learner interest.

## Adaptive Learning Engine:

## Description: This module adjusts the difficulty and type of content presented based on the learner's progress and performance.

## Features: Real-time data analysis, personalized learning paths, and feedback mechanisms to ensure each child receives appropriate challenges.

## Reward System:

## Description: This module provides motivational incentives to keep learners engaged and motivated through scores, badges, and other rewards.

## Features: Customizable rewards, progress milestones, and gamification elements that encourage continuous learning.

## Progress Tracking and Analytics:

## Description: This module tracks the child's learning progress and provides detailed analytics for educators and parents.

## Features: Real-time monitoring, comprehensive reports, and insights into individual learning patterns and areas needing improvement.

## User Feedback and Continuous Improvement:

## Description: This module collects feedback from users to continuously enhance the application’s functionality and effectiveness.

## Features: Feedback forms, regular updates, and improvements based on user suggestions and emerging educational research.

## Parental and Educator Dashboard:

## Description: This module provides a dedicated interface for parents and educators to manage and monitor the child's learning activities.

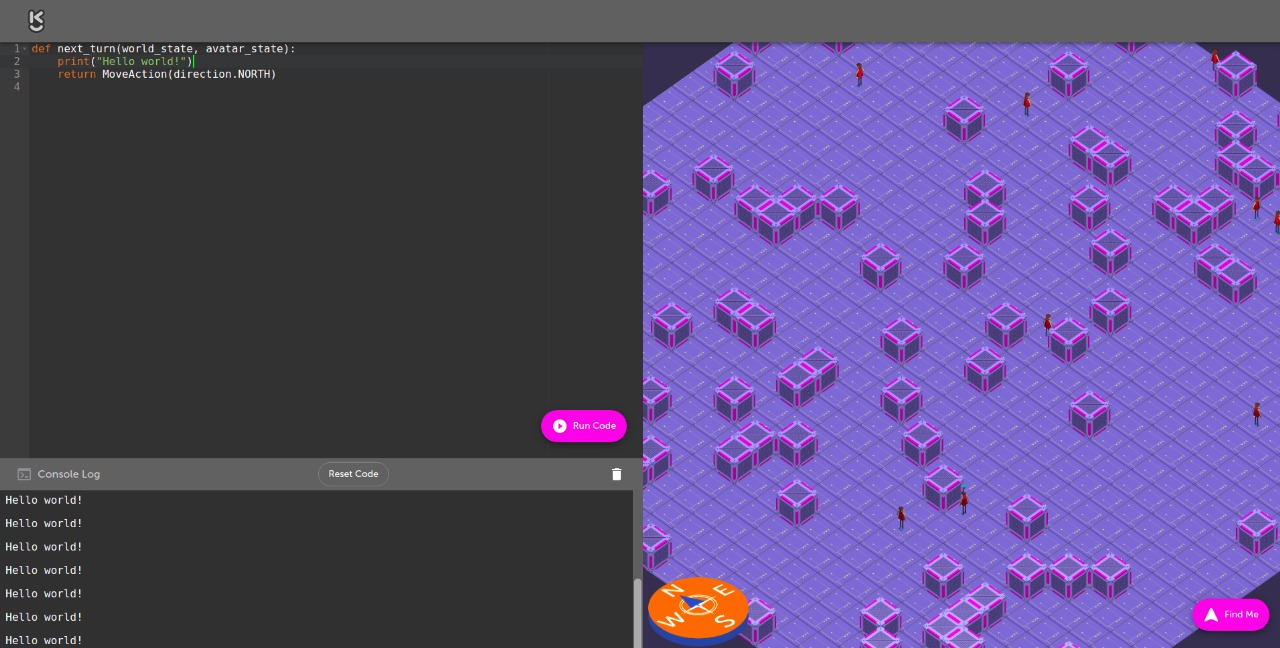
## Features: User-friendly dashboard, access to detailed progress reports, ability to set learning goals and monitor achievement, and communication tools for coordinating with other educators and caregivers.

## CHAPTER 5

**RESULTS AND DISCUSSIONS**

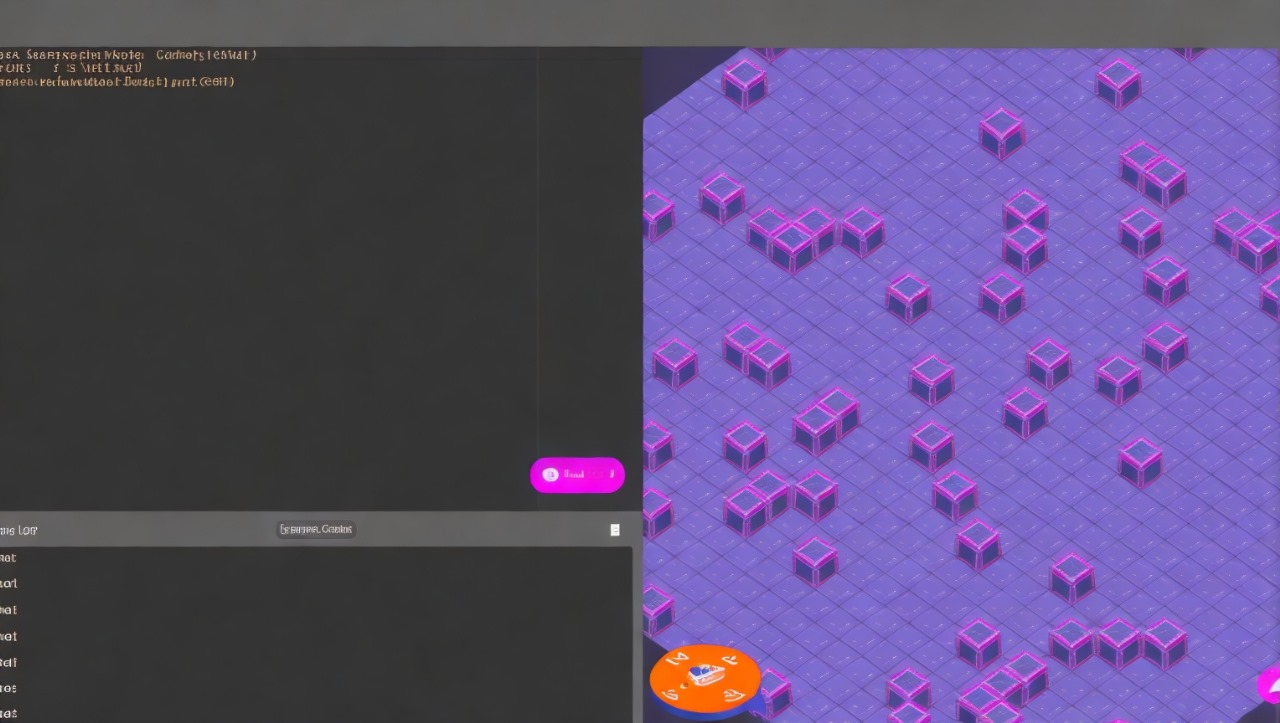
## OUTPUT

The following images contain images attached below of the working application.



## Fig 5.1: Output

**STEP 2: SIGNING INTO THE MAIN PAGE**

****

## Fig 5.1: Output



## Fig 5.1: Output

**5.2 SOURCE CODE:**

## SET UP.py

## -- coding: utf-8 --

## import re

## import sys

## from setuptools import find\_packages, setup

## with open("aimmo/\_init\_.py", "r") as fd:

## version = re.search(

## r'^\_version\_\s\*=\s\*[\'"]([^\'"]\*)[\'"]', fd.read(), re.MULTILINE

## ).group(1)

## try:

## from semantic\_release import setup\_hook

## setup\_hook(sys.argv)

## except ImportError:

## pass

## setup(

## name="aimmo",

## packages=find\_packages(exclude=[".tests", ".tests.\*"]),

## package\_dir={"aimmo": "aimmo"},

## include\_package\_data=True,

## install\_requires=[

## "cfl-common",

## "django==3.2.25",

## "django-csp==3.7",

## "django-js-reverse==0.9.1",

## "djangorestframework==3.13.1",

## "eventlet==0.31.0",

## "hypothesis==5.41.3",

## "kubernetes==26.1.0",

## "requests==2.31.0",

## ],

## tests\_require=["docker >= 3.5, < 3.6", "PyYAML == 5.4"],

## version=version,

## zip\_safe=False,

## from \_future\_ import print\_function

## from enum import Enum

## import re

## import sys

## import platform

## import subprocess

## import traceback

## import inspect

## from subprocess import PIPE, CalledProcessError

## # python2 support

## try:

## input = raw\_input

## except NameError:

## pass

## MINIKUBE\_VERSION = "latest"

## KUBECTL\_VERSION = "latest"

## class OSType(Enum):

## MAC = 1

## LINUX = 2

## WINDOWS = 3

## class ArchType(Enum):

## AMD64 = 1

## ARM64 = 2

## def main():

## try:

## os\_type = get\_os\_type()

## arch\_type = get\_arch\_type()

## setup = setup\_factory(os\_type, arch\_type)

## try:

## print("Starting setup for OS: %s\n" % os\_type.name)

## setup(os\_type, arch\_type)

## print("\nFinished setup.")

## except CalledProcessError as e:

## print("Something has gone wrong.")

## print("Command '%s' returned exit code '%s'" % (e.cmd, e.returncode))

## traceback.print\_exc()

## except OSError as e:

## print("Tried to execute a command that didn't exist.")

## traceback.print\_exc()

## except ValueError as e:

## print("Tried to execute a command with invalid arguments.")

## traceback.print\_exc()

## except KeyError as e:

## print("Setup encountered an error: %s" % e.args[0])

## except:

## print("An unexpected error has occured:\n")

## raise

## def get\_os\_type():

## """

## Return the OS type if one can be determined

## Returns:

## OSType: OS type

## """

## system = platform.system()

## system\_os\_type\_map = {

## "Darwin": OSType.MAC,

## "Linux": OSType.LINUX,

## "Windows": OSType.WINDOWS,

## }

## try:

## return system\_os\_type\_map[system]

## except KeyError:

## raise KeyError("'%s' system is not supported" % system)

## def get\_arch\_type():

## """

## Return the architecture type

## Returns:

## ArchType: architecture type

## """

## arch = platform.machine()

## arch\_type\_map = {

## "amd64": ArchType.AMD64,

## "x86\_64": ArchType.AMD64,

## "arm64": ArchType.ARM64,

## }

## try:

## return arch\_type\_map[arch]

## except KeyError:

## raise KeyError("'%s' architecture is not supported" % arch)

## def setup\_factory(os\_type, arch\_type):

## """

## Return the setup function which matches supplied host type

## Args:

## os\_type (OSType): the type of host to setup

## arch\_type (ArchType): host architecture type

## Returns:

## Callable: setup function

## """

## if os\_type == OSType.MAC:

## return mac\_setup

## elif os\_type == OSType.LINUX:

## return linux\_setup

## elif os\_type == OSType.WINDOWS:

## return windows\_setup

## raise RuntimeError("could not find setup function for supplied host type")

## def mac\_setup(os\_type, arch\_type):

## """

## Runs the commands needed in order to set up Kurono for MAC

## Args:

## os\_type (OSType): host OS type

## arch\_type (ArchType): host architecture type

## """

## tasks = [

## ensure\_homebrew\_installed,

## install\_sqlite3,

## install\_nodejs,

## install\_yarn,

## set\_up\_frontend\_dependencies,

## install\_pipenv,

## build\_pipenv\_virtualenv,

## install\_docker,

## install\_minikube,

## install\_kubectl,

## install\_helm,

## helm\_add\_agones\_repo,

## minikube\_start\_profile,

## helm\_install\_aimmo,

## ]

## \_create\_sudo\_timestamp()

## for task in tasks:

## task(os\_type, arch\_type)

## def windows\_setup(os\_type, arch\_type):

## raise NotImplementedError

## def linux\_setup(os\_type, arch\_type):

## """

## Runs the commands needed in order to set up Kurono for LINUX

## Args:

## os\_type (OSType): host OS type

## arch\_type (ArchType): host architecture type

## """

## tasks = [

## update\_apt\_packages,

## install\_nodejs,

## check\_for\_cmdtest,

## configure\_yarn\_repo,

## install\_yarn,

## install\_pip,

## install\_pipenv,

## build\_pipenv\_virtualenv,

## set\_up\_frontend\_dependencies,

## install\_docker,

## install\_minikube,

## install\_kubectl,

## install\_helm,

## helm\_add\_agones\_repo,

## minikube\_start\_profile,

## helm\_install\_aimmo,

## ]

## \_create\_sudo\_timestamp()

## for task in tasks:

## task(os\_type, arch\_type)

## def \_create\_sudo\_timestamp():

## """

## Request sudo access to create timestamp file for duration of setup

## """

## print("\033[1mrequesting\_sudo\_access\033[0m... ")

## # Request sudo password before task

## subprocess.Popen("sudo true", stdin=PIPE, stdout=PIPE, stderr=PIPE, shell=True).communicate()

## print("\033[1mrequesting\_sudo\_access\033[0m... [ \033[92mOK\033[0m ]")

## def \_cmd(command, comment=None):

## """

## Run command inside a terminal

## Args:

## command (str): command to be run

## comment (str): optional comment

## Returns:

## Tuple[int, List[str]]: return code, stdout lines output

## """

## stdout\_lines = []

## if not comment:

## # Set comment to calling function name

## comment = inspect.currentframe().f\_back.f\_code.co\_name

## if comment:

## print(" " \* 110, end="\r")

## print("\033[1m%s\033[0m...\n" % comment, end="\r")

## p = subprocess.Popen(command, stdin=PIPE, stdout=PIPE, stderr=PIPE, shell=True)

## for line in iter(p.stdout.readline, b""):

## stdout\_lines.append(line.decode("utf-8"))

## sys.stdout.write("%s\r" % line.decode("utf-8")[:-1].rstrip())

## sys.stdout.flush()

## # Delete line

## sys.stdout.write("\x1b[2K")

## sys.stdout.write("\x1b[1A")

## p.communicate()

## if p.returncode != 0:

## if comment:

## sys.stdout.write("\033[1m%s\033[0m... [ \033[93mFAILED\033[0m ]\n" % comment)

## for line in stdout\_lines:

## sys.stdout.write(f"{line}\n")

## raise CalledProcessError(p.returncode, command)

## if comment:

## sys.stdout.write("\033[1m%s\033[0m... [ \033[92mOK\033[0m ]\n" % comment)

## return (p.returncode, stdout\_lines)

## def ensure\_homebrew\_installed(os\_type, arch\_type):

## if os\_type == OSType.MAC:

## \_cmd("brew -v")

## def install\_sqlite3(os\_type, arch\_type):

## if os\_type == OSType.MAC:

## try:

## if \_cmd("sqlite3 -version", "check\_sqlite3")[0] == 0:

## return

## except CalledProcessError:

## pass

## \_cmd("brew install sqlite3")

## def install\_yarn(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## if \_cmd("yarn --version ", "check\_yarn")[0] == 0:

## return

## except CalledProcessError:

## pass

## if os\_type == OSType.MAC:

## \_cmd("npm install --global yarn", "install yarn")

## elif os\_type == OSType.LINUX:

## \_cmd("sudo npm install --global yarn", "install yarn")

## def set\_up\_frontend\_dependencies(os\_type, arch\_type):

## if os\_type == OSType.MAC:

## \_cmd("cd ./game\_frontend && yarn")

## elif os\_type == OSType.LINUX:

## \_cmd("cd ./game\_frontend && sudo yarn")

## def install\_pipenv(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## if \_cmd("pipenv --version", "check\_pipenv")[0] == 0:

## return

## except CalledProcessError:

## pass

## if os\_type == OSType.MAC:

## \_cmd("brew install pipenv")

## elif os\_type == OSType.LINUX:

## \_cmd("pip install pipenv")

## def build\_pipenv\_virtualenv(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## \_cmd("pipenv install --dev")

## def install\_docker(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## if \_cmd("docker -v", "check\_docker")[0] == 0:

## return

## except CalledProcessError:

## pass

## if os\_type == OSType.MAC:

## \_cmd("brew install --cask docker")

## elif os\_type == OSType.LINUX:

## # First time install needs to setup a repository

## # Update the package and install them

## # Add Docker's GPG key

## # The following command is used to setup the stable repository

## # Install docker

## docker\_install = """sudo apt-get update

## sudo apt-get install -y ca-certificates curl gnupg lsb-release

## curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

## echo \

## "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

## $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

## sudo apt-get update

## sudo apt-get install -y docker-ce

## sudo apt-get install -y docker-ce-cli

## sudo apt-get install -y containerd.io

## """

## try:

## \_cmd(docker\_install)

## except CalledProcessError:

## print("\nInstalation failed, trying again..\n")

## \_cmd(docker\_install)

## def install\_minikube(os\_type, arch\_type, version=MINIKUBE\_VERSION):

## comment = "install\_minikube"

## if version == "latest":

## rc, lines = \_cmd("curl https://api.github.com/repos/kubernetes/minikube/releases/latest | grep tag\_name")

## match = re.search(r"(v[0-9\.]+)", lines[0])

## if rc == 0 and match:

## version = match.group(1)

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## \_, lines = \_cmd("minikube version", "check\_minikube")

## if version in lines[0]:

## return

## except CalledProcessError:

## pass

## if os\_type == OSType.MAC:

## \_cmd(

## "curl -Lo minikube https://storage.googleapis.com/minikube/releases/%s/minikube-darwin-%s"

## % (version, arch\_type.name.lower()),

## comment + ": download",

## )

## elif os\_type == OSType.LINUX:

## \_cmd(

## "curl -Lo minikube https://storage.googleapis.com/minikube/releases/%s/minikube-linux-%s"

## % (version, arch\_type.name.lower()),

## comment + ": download",

## )

## if os\_type in [OSType.MAC, OSType.LINUX]:

## \_cmd("chmod +x minikube", comment + ": set permissions")

## \_cmd("sudo mv minikube /usr/local/bin/", comment + ": copy binary")

## def install\_kubectl(os\_type, arch\_type, version=KUBECTL\_VERSION):

## comment = "install\_kubectl"

## if version == "latest":

## rc, lines = \_cmd("curl -L -s https://dl.k8s.io/release/stable.txt")

## if rc == 0:

## version = lines[0]

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## \_, lines = \_cmd("kubectl version --client --short", "check\_kubectl")

## if version in lines[0]:

## return

## except CalledProcessError:

## pass

## if os\_type == OSType.MAC:

## \_cmd(

## "curl -Lo kubectl https://dl.k8s.io/release/%s/bin/darwin/%s/kubectl"

## % (

## version,

## (arch\_type.name).lower(),

## ),

## comment + ": download",

## )

## if os\_type == OSType.LINUX:

## \_cmd(

## "curl -Lo kubectl https://dl.k8s.io/release/%s/bin/linux/%s/kubectl"

## % (

## version,

## (arch\_type.name).lower(),

## ),

## comment + ": download",

## )

## if os\_type in [OSType.MAC, OSType.LINUX]:

## \_cmd("chmod +x kubectl", comment + ": set permissions")

## \_cmd("sudo mv kubectl /usr/local/bin/", comment + ": copy binary")

## def install\_helm(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## rc, \_ = \_cmd("helm version > /dev/null", "check\_helm")

## if rc == 0:

## return

## except CalledProcessError:

## pass

## \_cmd("curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash")

## def helm\_add\_agones\_repo(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## \_cmd("helm repo add agones https://agones.dev/chart/stable && " "helm repo update")

## def minikube\_start\_profile(os\_type, arch\_type):

## if os\_type == OSType.MAC:

## \_cmd("minikube start -p agones --driver=hyperkit")

## if os\_type == OSType.LINUX:

## \_cmd("minikube start -p agones")

## def helm\_install\_aimmo(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## if \_cmd("helm status -n agones-system aimmo > /dev/null", "check\_helm\_aimmo")[0] == 0:

## return

## except CalledProcessError:

## pass

## \_cmd(

## "minikube profile agones && "

## "helm install aimmo --namespace agones-system --create-namespace agones/agones"

## )

## def install\_pip(os\_type, arch\_type):

## if os\_type == OSType.LINUX:

## try:

## if \_cmd("pip --version", "check\_pip")[0] == 0:

## return

## except CalledProcessError:

## pass

## \_cmd("sudo apt-get install python3-pip", "install\_pip")

## def install\_nodejs(os\_type, arch\_type):

## if os\_type in [OSType.MAC, OSType.LINUX]:

## try:

## if \_cmd("node --version", "check\_nodejs")[0] == 0:

## return

## except CalledProcessError:

## pass

## if os\_type == OSType.MAC:

## \_cmd("brew install node@14")

## if os\_type == OSType.LINUX:

## \_cmd("curl -fsSL https://deb.nodesource.com/setup\_14.x | sudo -E bash -" "sudo apt-get install -y nodejs")

## def check\_for\_cmdtest(os\_type, arch\_type):

## """

## This function is for use within the Linux setup section of the script. It checks if

## the cmdtest package is installed, if it is we ask the user if we can remove it, if yes

## we remove the package, if not the process continues without removing it.

## """

## if os\_type == OSType.LINUX:

## try:

## \_cmd("dpkg-query -W -f='{status}' cmdtest")

## except CalledProcessError:

## return

## while True:

## choice = input(

## "Looks like cmdtest is installed on your machine. "

## "cmdtest clashes with yarn so we recommend to remove it. "

## "Is it okay to remove cmdtest? [y/n]"

## ).lower()

## if choice in ["y", "yes"]:

## \_cmd("sudo apt-get remove -y cmdtest", "remove\_cmdtest")

## break

## if choice in ["n", "no"]:

## print("Continuing without removing cmdtest...")

## break

## print("Please answer 'yes' or 'no' ('y' or 'n').")

## def update\_apt\_packages(os\_type, arch\_type):

## if os\_type == OSType.LINUX:

## \_cmd("sudo apt-get update")

## def configure\_yarn\_repo(os\_type, arch\_type):

## comment = "configure\_yarn"

## if os\_type == OSType.LINUX:

## \_cmd(

## "curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -",

## comment + ": add key",

## )

## \_cmd(

## 'echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list',

## comment + ": add repo",

## )

## if \_name\_ == "\_main\_":

## main()

## [7:08 PM, 5/19/2024] Aaron Joel Cse Rec: coverage:

## precision: 2

## round: down

## range: "70...100"

## status:

## patch:

## default:

## target: 90%

## ignore:

## - "aimmo\_setup.py"

## - "run.py"

## - "aimmo/static/\*"

## - "aimmo/static/\*/"

## - "aimmo/static/\*//"

## - "aimmo-game/agones/\*"

## - "aimmo-game/agones/\*/"

## - "aimmo\_runner/\*"

## - "aimmo\_runner/\*/"

## - "test\_utils/\*"

## - "setup.py"

## comment: false

## #!/usr/bin/env python

## import argparse

## import logging

## import traceback

## from aimmo\_runner import runner

## logging.basicConfig()

## parser = argparse.ArgumentParser(description="Runs Kurono.")

## parser.add\_argument(

## "-t",

## "--target",

## dest="build\_target",

## choices=["runner", "tester"],

## action="store",

## default="runner",

## help="""Specify the build stage you wish the docker containers to stop at.

## By default we simply run the project. This can be used to run the tests

## but it is recommended that you used 'all\_tests.py'

## Options: runner, tester """,

## )

## parser.add\_argument(

## "-c",

## "--using-cypress",

## dest="using\_cypress",

## action="store\_true",

## default=False,

## help="""Specify if you want to run the project for running Cypress tests. This

## disables the building of the Docker images and builds the frontend in production

## mode without watching for changes.""",

## )

## if \_name\_ == "\_main\_":

## try:

## args = parser.parse\_args()

## runner.run(

## using\_cypress=args.using\_cypress,

## build\_target=args.build\_target,

## )

## except Exception as err:

## traceback.print\_exc()

## raise

## import atexit

## import os

## import platform

## import subprocess

## import kubernetes

## from kubernetes.client import AppsV1Api, CoreV1Api

## from kubernetes.config import load\_kube\_config

## from .docker\_scripts import build\_docker\_images

## from .shell\_api import run\_command

## MINIKUBE\_EXECUTABLE = "minikube"

## def get\_ip():

## internal\_ip = str(

## run\_command(

## "minikube -p agones ssh grep host.minikube.internal /etc/hosts".split(),

## capture\_output=True,

## ).split()[0],

## "utf-8",

## )

## return internal\_ip

## def delete\_components():

## apps\_api\_instance = AppsV1Api()

## api = CoreV1Api()

## for rs in apps\_api\_instance.list\_namespaced\_deployment("default").items:

## apps\_api\_instance.delete\_namespaced\_deployment(

## body=kubernetes.client.V1DeleteOptions(),

## name=rs.metadata.name,

## namespace="default",

## grace\_period\_seconds=0,

## )

## for service in api.list\_namespaced\_service(namespace="default").items:

## api.delete\_namespaced\_service(service.metadata.name, "default")

## delete\_fleet\_on\_exit()

## def restart\_pods():

## """

## Disables all the components running in the cluster and starts them again

## with fresh updated state.

## :param game\_creator\_yaml: Replication controller yaml settings file.

## """

## print("Restarting pods")

## try:

## run\_command(["kubectl", "create", "-f", "agones/fleet.yml"])

## except subprocess.CalledProcessError:

## run\_command("kubectl delete fleet aimmo-game --ignore-not-found".split(" "))

## run\_command("kubectl delete --all deployment -n default".split(" "))

## run\_command(["kubectl", "create", "-f", "agones/fleet.yml"])

## def create\_roles():

## """

## Applies the service accounts, roles, and bindings for restricting

## the rights of certain pods and their processses.

## """

## run\_command(["kubectl", "apply", "-Rf", "rbac"])

## def delete\_fleet\_on\_exit():

## print("Exiting")

## print("Deleting aimmo-game fleet")

## run\_command(

## [

## "kubectl",

## "delete",

## "fleet",

## "aimmo-game",

## "--ignore-not-found",

## ]

## )

## def start(build\_target=None):

## """

## The entry point to the minikube class. Sends calls appropriately to set

## up minikube.

## """

## if platform.machine().lower() not in ("amd64", "x86\_64"):

## raise ValueError("Requires 64-bit")

## os.environ["MINIKUBE\_PATH"] = MINIKUBE\_EXECUTABLE

## # We assume the minikube was started with a profile called "agones"

## load\_kube\_config(context="agones")

## create\_roles()

## build\_docker\_images(MINIKUBE\_EXECUTABLE, build\_target=build\_target)

## restart\_pods()

## atexit.register(delete\_components)

## print("Cluster ready")

## [7:08 PM, 5/19/2024] Aaron Joel Cse Rec: from \_future\_ import absolute\_import

## import logging

## import os

## import sys

## import django

## from django.conf import settings

## from .shell\_api import log, run\_command, run\_command\_async

## ROOT\_DIR\_LOCATION = os.path.abspath(os.path.dirname((os.path.dirname(\_file\_))))

## \_MANAGE\_PY = os.path.join(ROOT\_DIR\_LOCATION, "example\_project", "manage.py")

## \_FRONTEND\_BUNDLER\_JS = os.path.join(ROOT\_DIR\_LOCATION, "game\_frontend", "djangoBundler.js")

## PROCESSES = []

## def create\_superuser\_if\_missing(username, password):

## from django.contrib.auth.models import User

## try:

## User.objects.get\_by\_natural\_key(username)

## except User.DoesNotExist:

## log("Creating superuser %s with password %s" % (username, password))

## User.objects.create\_superuser(username=username, email="admin@admin.com", password=password)

## def build\_worker\_package():

## run\_command([os.path.join(ROOT\_DIR\_LOCATION, "aimmo\_runner", "build\_worker\_wheel.sh")], capture\_output=True)

## def build\_frontend(using\_cypress, capture\_output):

## if using\_cypress:

## run\_command(["node", \_FRONTEND\_BUNDLER\_JS], capture\_output=capture\_output)

## else:

## frontend\_bundler = run\_command\_async(["node", \_FRONTEND\_BUNDLER\_JS], capture\_output=capture\_output)

## PROCESSES.append(frontend\_bundler)

## def start\_game\_servers(build\_target, server\_args, capture\_output: bool):

## parent\_dir = os.path.dirname(os.path.dirname(os.path.abspath(\_file\_)))

## sys.path.append(os.path.join(parent\_dir, "aimmo\_runner"))

## os.chdir(ROOT\_DIR\_LOCATION)

## # Import minikube here, so we can install the dependencies first

## from aimmo\_runner import minikube

## minikube.start(build\_target=build\_target)

## server\_args.append("0.0.0.0:8000")

## os.environ["AIMMO\_MODE"] = "minikube"

## run\_command(

## ["python", \_MANAGE\_PY, "start\_game\_servers\_for\_running\_games"],

## capture\_output=capture\_output,

## )

## def run(server\_wait=True, using\_cypress=False, capture\_output=False, test\_env=False, build\_target=None):

## logging.basicConfig()

## build\_worker\_package()

## if test\_env:

## os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "test\_settings")

## else:

## sys.path.insert(0, os.path.join(ROOT\_DIR\_LOCATION, "example\_project"))

## os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "settings")

## django.setup()

## if using\_cypress:

## settings.DEBUG = False

## os.environ["LOAD\_KUBE\_CONFIG"] = "0"

## os.environ["NODE\_ENV"] = "development" if settings.DEBUG else "production"

## build\_frontend(using\_cypress, capture\_output)

## run\_command(["pip", "install", "-e", ROOT\_DIR\_LOCATION], capture\_output=capture\_output)

## if not test\_env:

## run\_command(["python", \_MANAGE\_PY, "migrate", "--noinput"], capture\_output=capture\_output)

## run\_command(["python", \_MANAGE\_PY, "collectstatic", "--noinput", "--clear"], capture\_output=capture\_output)

## server\_args = []

## if not using\_cypress:

## start\_game\_servers(build\_target, server\_args, capture\_output)

## os.environ["SERVER\_ENV"] = "local"

## server = run\_command\_async(["python", \_MANAGE\_PY, "runserver"] + server\_args, capture\_output=capture\_output)

## PROCESSES.append(server)

## if server\_wait:

## try:

## game.wait()

## except NameError:

## pass

## server.wait()

## return PROCESSES

## coding: utf-8 --

## from setuptools import find\_packages, setup

## setup(

## name="aimmo\_runner",

## packages=find\_packages(),

## include\_package\_data=True,

## install\_requires=[

## "django==3.2.25",

## "django-forms-bootstrap",

## "django-js-reverse",

## "docker<6",

## "eventlet",

## "hypothesis",

## "kubernetes==26.1.0",

## "psutil",

## "requests",

## "six",

## ],

## zip\_safe=False,

## )

## Import errno

## import os

## import platform

## import stat

## import subprocess

## import sys

## from subprocess import CalledProcessError

## try:

## from urllib.request import urlretrieve, urlopen

## except ImportError:

## from urllib import urlretrieve, urlopen

## BASE\_DIR = os.path.abspath(os.path.dirname(os.path.dirname(\_file\_)))

## TEST\_BIN = os.path.join(BASE\_DIR, "test-bin")

## OS = platform.system().lower()

## FILE\_SUFFIX = ".exe" if OS == "windows" else ""

## KUBECTL = os.path.join(TEST\_BIN, "kubectl%s" % FILE\_SUFFIX)

## MINIKUBE = os.path.join(TEST\_BIN, "minikube%s" % FILE\_SUFFIX)

## FNULL = open(os.devnull, "w")

## def log(message):

## sys.stderr.write(message + "\n")

## def run\_command(args, capture\_output=False):

## try:

## if capture\_output:

## return subprocess.check\_output(args)

## else:

## subprocess.check\_call(args)

## except CalledProcessError as e:

## log("Command failed with exit status %d: %s" % (e.returncode, " ".join(args)))

## raise

## def run\_command\_async(args, capture\_output=False):

## env = os.environ.copy()

## if capture\_output is True:

## p = subprocess.Popen(args, stdout=FNULL, stderr=subprocess.STDOUT, env=env)

## else:

## p = subprocess.Popen(args, env=env)

## return p

## def binary\_exists(filename):

## # Check if binary is callable on our path

## try:

## run\_command([filename], True)

## return True

## except OSError:

## return False

## def download\_exec(url, dest):

## dest = urlretrieve(url, dest)[0]

## make\_exec(dest)

## def make\_exec(file):

## current\_stat = os.stat(file)

## os.chmod(file, current\_stat.st\_mode | stat.S\_IEXEC)

## def get\_latest\_github\_version(repo):

## result = urlopen("https://github.com/%s/releases/latest" % repo)

## return result.geturl().split("/")[-1]

## [7:09 PM, 5/19/2024] Aaron Joel Cse Rec: from \_future\_ import absolute\_import

## from django.contrib import admin

## from .models import Avatar, Game

## class GameDataAdmin(admin.ModelAdmin):

## search\_fields = ["id", "owner\_username", "owner\_email"]

## list\_display = ["id", "owner", "game\_class", "school", "worksheet\_id", "status", "creation\_time", "is\_archived"]

## raw\_id\_fields = ["owner", "main\_user", "can\_play", "game\_class"]

## readonly\_fields = ["players", "auth\_token"]

## def players(self, obj):

## teacher\_user = obj.game\_class.teacher.new\_user

## players = f"{teacher\_user.first\_name} {teacher\_user.last\_name}\n"

## players += "\n".join([student.new\_user.first\_name for student in obj.game\_class.students.all()])

## return players

## def school(self, obj):

## if obj.game\_class:

## return obj.game\_class.teacher.school

## else:

## return None

## def stop\_game(game\_admin, request, queryset):

## for game in queryset:

## game.status = Game.STOPPED

## game.save()

## stop\_game.short\_description = "Stop selected games"

## class AvatarDataAdmin(admin.ModelAdmin):

## search\_fields = ["owner\_username", "owner\_email"]

## list\_display = ["id", "owner\_name", "game\_id"]

## raw\_id\_fields = ["game"]

## readonly\_fields = ["owner", "auth\_token"]

## def owner\_name(self, obj):

## return obj.owner

## def game\_id(self, obj):

## return obj.game

## admin.site.register(Game, GameDataAdmin)

## admin.site.register(Avatar, AvatarDataAdmin)

## GameDataAdmin.actions.append(stop\_game)

## from django.conf import settings

## #: URL function for locating the game server, takes one parameter game

## GAME\_SERVER\_URL\_FUNCTION = getattr(settings, "AIMMO\_GAME\_SERVER\_URL\_FUNCTION", None)

## GAME\_SERVER\_PORT\_FUNCTION = getattr(settings, "AIMMO\_GAME\_SERVER\_PORT\_FUNCTION", None)

## GAME\_SERVER\_SSL\_FLAG = getattr(settings, "AIMMO\_GAME\_SERVER\_SSL\_FLAG", False)

## # Hostname for django server to pass onto a game server

## DJANGO\_BASE\_URL\_FOR\_GAME\_SERVER = getattr(settings, "AIMMO\_DJANGO\_BASE\_URL", "localhost")

## CHAPTER 6

**CONCLUSION AND FUTURE ENHANCEMENT**

## 6.1 CONCLUSION

The development of a gamified learning application tailored for dyslexic children marks a significant advancement in educational technology. Dyslexia, affecting a substantial portion of the population, poses unique challenges in reading, writing, and spelling that traditional educational methods often fail to address adequately. This project has demonstrated that integrating the principles of game design with adaptive learning technologies can create an engaging, effective, and personalized educational experience meeting the specific needs of dyslexic learners.

The interactive learning games within the application target critical areas such as phonemic awareness, reading fluency, and spelling. By making these games adaptive, the application ensures each child receives a tailored learning experience adjusting to their individual pace and progress. This adaptability is crucial for maintaining the child's engagement and motivation, essential for effective learning. The inclusion of a reward system further enhances motivation by providing positive reinforcement through scores, badges, and other incentives, making learning enjoyable and encouraging continuous effort and progress.

The robust progress tracking and analytics module offers educators and parents valuable insights into the child's learning journey. Real-time monitoring and detailed reports enable informed decision-making and targeted interventions, ensuring each child receives the support they need to overcome their specific challenges. The user feedback and continuous improvement module ensures the application evolves in response to user needs and emerging educational research, maintaining its relevance and effectiveness.

Looking ahead, several potential enhancements could further improve the functionality and impact of the gamified learning application. Integration of advanced technologies such as artificial intelligence and machine learning could enhance adaptiveness and personalized learning experiences. Expanding content to cover broader subject areas and adding multi-language support could make the application accessible to a wider audience. Enhanced parental and educator involvement, usability and accessibility improvements, and robust data privacy and security measures are also essential considerations for future development.

In conclusion, the gamified learning application for dyslexic children has the potential to transform the educational experience, providing tailored support and resources to empower every child to succeed academically and build confidence in their abilities. Through continued evolution and improvement, we can make significant strides in addressing the educational challenges of dyslexia and ensuring every child reaches their full potential.

## REFERENCES

* <https://www.wikipedia.org/>
* [Dyslexia in Schools: Understanding and Teaching Students with Dyslexia - Waterford.org](https://www.waterford.org/education/dyslexia-in-schools/)
* <https://www.github.com/>
* <https://www.simplilearn.com/>
* https://www.readingresource.net/