



IIT Bombay

PROJECT REPORT

E3STORE – MATHS AND BRAIN ASSESSMENT TOOLSET

Submitted in Fulfillment of Summer Internship Project.

TABLE OF CONTENTS

Contents

Project Report Approval	1
Acknowledgement	2
Abstract	3
1. Introduction	4
2. Learning Tool Interoperability	7
3. Project Stakeholders	8
4. Use Cases	13
5. Learning Content	29
6. Functional Requirements	22
7. Non-Functional Requirements	28
8. Brief Module Description	Error! Bookmark not defined.
9. Module Interaction Diagram	Error! Bookmark not defined.
10. Modules	Error! Bookmark not defined.
11. Testing	64
12. User Manual - Game Engine	79
13. User Manual - Platform	92
14. Challenges	104
15. Response and Feedback	106
16. References	110

Project Report Approval

The project Report of E3Store – Math and Brain Assessment Toolset is approved in response to the successful completion of Summer Internship 2016 for the following interns:

1. Nandan Sukthankar
2. Manveer Singh
3. Sunny Mazumder
4. Aditya N.K.
5. Utkarsh Jain
6. Shefali Gupta
7. Saurav Kumar
8. Mayank Arya
9. Shubham Shah
10. Srinidhi Bhat

(Prof. (Dr.) D.B. Phatak)
Guide

(Mr. Kishor Chaturani,
OSS Enterprise Architect,
TCS Ltd.)
Project Mentor

(Mr. Avinash Awate
Program Director, MOOCs Lab, IITB)
Supervisor

(Ms. Surabhi Baoker,
Solution Architect, TCS Ltd.)
Project Mentor

Acknowledgement

We, the summer interns team of '**E3Store-Math and Brain Assessment Toolset**' are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put our ideas and assigned work, well above the level of simplicity and into something concrete.

We all thank whole heartedly to,

- **Prof. (Dr.) D.B. Phatak** for selecting us as a part of his valuable project.
- Our Supervisor **Mr. Avinash Awate** for his valuable support and guidance.
- Our mentors **Mr. Kishor Chaturani** and **Ms. Surabhi Baoker** for their guidance. Due to sharing of their experiences of real world situations, we were able to manage and complete the project smoothly.
- **Powai English High School, Powai** for providing us the opportunity to show a demo of our platform to their kids and record their feedback.
- **IITB Administration** for making our stay here at IITB memorable and pleasant.

Abstract

While India has made great strides in improving its education system, much still remains to be done. Education always holds the key towards success. The problems in Indian society today are a variation in the quality of education, dependent upon the wealth, or lack thereof, in particular areas. And particularly mathematics and logical reasoning has been an area where the case is too pathetic. Due to an augment in job opportunities, teaching has become a secondary interest, consequently, lesser student participation at schools at least at primary and secondary level. Our team decided to bend the rules of traditional teaching, and provide students of age group 4-9Yrs., with means to learn mathematics and logic without a physical teacher. Since computers have become one of the biggest allies, we decided to bridge the gap between the student and the teacher. The student, plays interactive games on our platform and learn the basics of mathematics and logic while the mentor (or the parent/guardian) gets a feedback of their progress in each subject. The mentor can also get an idea of the student's progress by a dynamic leaderboard which projects the score of all the students for each game. We envisage a platform that encourages children to learn on their own. This platform can even be used by teachers to make new games, thereby providing a flexible methodology for teaching.

1. Introduction

1.1 Current State of Online Learning

Massive Open Online Courses (MOOCs) is overpowering the area of online education everywhere. Professors from world class institutes are making online content (and assessments) available both in free and (increasingly) paid courses. Renowned institutions are accepting these credits as part of their curriculum.

In the US, school education is undergoing a transformation. More and more schools are adopting hybrid models of learning over the traditional classroom based (rote) approach.

Assessment of learning is a weak area especially in MOOCs where physical assessment is impossible. In the hybrid models as well, physical assessments (as against the e-Assessments) bring in a subjective element. An e-Assessment has the risk of large scale 'copying' and hence has to be properly designed.

1.2 The future of learning in the world

Our challenge isn't making incremental improvements in to an education model designed in 1893. Our opportunity and our obligation is to reimagine our schools and give all kids an education that will help them thrive in a world that values them for what they can do, not the facts that they know.

We believe that the present situation is only a transitory phenomenon. Learning is all set for a revolutionary change.

We believe real student centric learning will finally arrive 'in a decade or so. Any individual (including a child) will be able to choose what he/she wishes to learn, what degree of mastery he/she wishes to achieve in the chosen field, when he/she wishes to acquire the knowledge and at what rate he/she is comfortable in acquiring the knowledge. There will be no certificate examinations like our X and XII board, instead each 'subject' will have its own definitions of mastery and its own certificates. Such a freedom to learn is now available in 'hobby' fields such as dancing, singing, cooking etc.

One of the key ingredients to facilitate such learning is absorption of technology in the process of learning. The current MOOCs attempt to bring the classroom to the learner's laptop. The next generation content will expand the learning to what is not possible in a classroom.

The other key ingredient is the use of technology to assess a learner's performance by automatic evaluation of learner assignments. By providing automatic assessment reports on completed tasks, the learner gets immediate feedback and can then resubmit the task. This will reduce the e-Mentoring requirement only for certain tasks.

1.3 Our Purpose

To allow student centric learning platform we need blended mode of education where teacher of the online course is also a

mentor for the student. The learning management systems should have content as well as assessments which are not just based on the subjective knowledge of the student.

With e3store we have tried to reach the unexplored subjects purposely choosing the non-traditional areas (such as music and video). It covers subjects like mathematics, physics, music and video assessment tools as well as brain developing activities.

1.4 Objective

The objective is to provide an interactive learning platform for kids.

1.5 Scope

This platform will target kids of age-group 4--9 years.

1.6 What is expected to achieve?

Using this platform kids will learn fundamental concepts and methodologies to solve basic mathematical, logical and reasoning problems.

1.7 How it will be achieved?

Through interactive and fun filled games, puzzles and audio/visual content that will be best suited for this age group.

2. Learning Tool Interoperability

- The ability for a user within a VLE (or other web-based system) to seamlessly access a separate learning application, an item of protected content, or other restricted resource.
- LTI has its origins in the IMS Tools Interoperability specifications released in 2006.
- LTI is essentially provided a means of connecting two systems together: a “Tool Consumer” which “consumes” the tool, and a “Tool Provider” which “provides” the Tool to be used in the Tool Consumer. A Tool Consumer would typically be an LMS.

When a user is passed from the VLE to the other system, the following data may be carried with them:

- Details about them (including name and email address);
- Details about the institutional context (such as the VLE being used);
- Details of the specific context from which they are coming (such as an online course); and
- Their role within that context (such as “teacher” or “learner”).
- This “launch” process occurs in a secure manner (using OAuth) via the user’s web browser. A connection between the two systems is created by simply entering the following details in the VLE:
 - The URL to the other system;
 - A value to identify the customer (known as the “consumer key”);
 - A shared secret to secure the connection.

3. Project Stakeholders

Project Stakeholder is an individual, group, or organization, who may affect, be affected by a decision, activity, or outcome of a project.

Stakeholders	Expectations of Stakeholders	Roles	Requirements to use the platform
Kids	To have fun.	To learn mathematical and logical concepts by playing games	Basic knowledge of computers/ smart phones
Parents	-Fun and entertaining: by including cartoons and their favorite characters. -Engaging: interactive interface. -Intuitive: adding blinkers to capture attention.	-Mentoring their children. -Assessing the progress.	-Basic knowledge of computer/ smartphones.

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

	<ul style="list-style-type: none"> -Simple: not many keys or action to perform. -Economical: lesser bandwidth requirement, data volume consumption, good speed, good connectivity. -Secured: no redirection to malicious sites or advertisements. 		
School/ Day Care Centres	<ul style="list-style-type: none"> -Minimum employees: few people should be required to handle the kids. -System requirements: Minimum hardware, less 	<ul style="list-style-type: none"> -Mentoring the children -Assessing the children 	<ul style="list-style-type: none"> -Adequate systems with minimum configuration to run the product.

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

	<p>consumption of data and bandwidth.</p> <p>-Concurrency: Multiple users able to access various content at same time.</p> <p>-Simultaneous result dispatch: Hand to hand mail/message to the parents at the finish of each game. Also feedback to parents on how they should improve day to day interaction with the kid.</p> <p>-Monetary benefits: proper advertising</p>		<p>-Provide a local admin to manage the product.</p>
Administrator (Global)	<p>-Bugs free and smooth</p>	<p>-To maintain and update the platform.</p>	<p>-Sound technical knowledge</p>

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

	<p>execution of the platform.</p> <ul style="list-style-type: none"> -Platform must be easy to maintain. 	<ul style="list-style-type: none"> -Troubleshoot in case of errors. 	<ul style="list-style-type: none"> -Good communication skills.
Administrator (Local)	<ul style="list-style-type: none"> -Bugs free and smooth execution of the platform. -Platform must be easy to maintain. 	<ul style="list-style-type: none"> -To report the bugs to the global admin. 	<ul style="list-style-type: none"> -Sound technical knowledge -Good communication skills.
Teacher	<ul style="list-style-type: none"> -Must help in completing the prescribed course on time. -Must enable a child to learn -Must be fun and engaging. -Intuitive: Must require minimum effort from teacher's side. 	<ul style="list-style-type: none"> -Assist the child to learn from the content provided on the platform. 	<ul style="list-style-type: none"> -Basic knowledge of computers/ smartphones.

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

<p>Executive Sponsor</p>	<ul style="list-style-type: none"> - Ensuring the project is on time, on budget and on scope. - Providing feedback on status reports, technical, reports, feasibility studies, specifications, prototypes. - Providing presentations, animations, videos, demonstrations - Providing final technical report and one-page summary 	<ul style="list-style-type: none"> - Helping with project matters such as funding, scope clarification, progress monitoring. - Involved in authorizing changes in scope, phase-end reviews. 	<ul style="list-style-type: none"> - Basic knowledge about the project.
---------------------------------	--	---	--

4. Use Cases

A use case is a list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal. The actor can be a human or other external system.

3.01 Individual User Registration

Use Case	Individual User Registration
Description	The platform must be able to successfully register the user.
Actor(s)	1) Parent 2) Kid
Preconditions	1) Working internet connection. 2) Platform must be running
Actions	1) User provides Name, DOB, Residential address, parent's email and contact info, gender, unique username and an 8 digit alphanumeric password and school name.
Post conditions	1) User will be registered and will be able to login.

3.02 Batch User Registration

Use Case	Batch User Registration
Description	The platform must be able to successfully register the users in bulk
Actor(s)	Local Administrator
Preconditions	1) Working internet connection. 2) Platform must be running
Actions	1) Administrator chooses to provide individual details of the user. 2) Administrator chooses to provide bulk data in the batch registration form.
Post conditions	1) All the users will be registered and will be able to login.

3.03 User Login

Use Case	User Login
Description	1) The registered user must be able to login into the platform. 2) In case, login fails, platform must display login failure notification.
Actor(s)	1) Kid
Pre-conditions	1) The platform must be opened. 2) User must be registered
Actions	1) User provides username and password

Post conditions	<ul style="list-style-type: none"> 1) User will be able to use his profile. 2) User will be able to use the content.
-----------------	--

3.04 Update User Profile

Use Case	Update User Profile
Description	1) Successfully update the profile
Actor(s)	<ul style="list-style-type: none"> 1) Parent 2) Kid 3) Local Admin
Expectation	1) Successfully update the profile
Pre-conditions	<ul style="list-style-type: none"> 1) The kid/local admin must be online 2) The kid/local admin must be logged in.
Actions	<ul style="list-style-type: none"> 1) Edit the School Name, Parent's contact number / email, residential address, password. 2) Edit the details of a user by the local admin based on the user ID of the kid.
Post conditions	1) User's Profile will be updated.

3.05 School Registration

Use Case	School Registration
Description	1) School will be registered to the platform. 2) Local Admin Registration
Actor(s)	1) Local Admin
Pre-conditions	1) Working Internet
Actions	1) Provides School Name, School Address & School Affiliation Number. 2) Provides Local Admin Name, email / contact info for local admin, Local Admin Image 3) Provides Principal Email Address, Contact Number
Post conditions	1) School will be registered. 2) Local Admin will be registered.

3.06 Update Batch

Use Case	Update batch
Description	1) In a batch, new users can be added and existing users can be removed.
Actor	1) Kid 2) Local Admin
Pre-conditions	1) The Local Admin must be logged in. 2) The Local Admin must be online.
Actions	1) New users is added to a batch. 2) Existing users can be removed from a batch.

Post conditions	1) Batch members are updated.
-----------------	-------------------------------

3.07 Selection of Content Category

The content category refers to the classification of the content i.e. Mathematics and Brain Games sections.

Use Case	Selection of Content Category
Description	1) Selected category content will be displayed.
Actors	1) Kid
Pre-condition	1) User must be logged in
Actions	1) The User will select the required category.
Post condition	1) Contents of the category will be available to use for the user.

3.08 Accessing Content

Use Case	Accessing Content
Description	1) Use the content in an interactive way
Actor(s)	1) Kid
Preconditions	1) User must have selected the content category
Actions	1) The user will be displayed a wide variety of content from where the user will be choosing from.

Post conditions	1) The user will be able to access the content.
-----------------	---

3.09 Updating Content

Use Case	Updating Content
Description	1) Required contents successfully updated
Actor(s)	1) Global Admin
Preconditions	1) Required content must be available. 2) The Global Admin must be Logged in.
Actions	1) Global Admin will add the content using the option available exclusive to the global admin.
Post conditions	1) The Updated contents will be available for use.

3.10 Feedback

Use Case	Feedback
Description	1) Feedback is reported successfully and suggestions to be analyzed.
Actor(s)	1) Kid 2) Parent 3) Local Admin

Pre-condition	1) Feedback Sender (User, Parent and Local Admin) must have gone through the content.
Actions	1) User can send the feedback at the given admin mail id in the platform
Post Condition	1) Feedback is reported successfully to the Admin.

3.11 Reporting

Use Case	Reporting
Description	1) Parent will receive the assessment report of the child through push notification on their email. 2) The parent will be able to see the child's relative performance on the leader board.
Actors	1) Parent
Precondition	1) The parent information must be registered. 2) The user is using the content actively.
Actions	1) Automatic assessment report will be generated and sent to the parents periodically. 2) The leader board will be dynamically updated. 3) The user profile will be dynamically updated with the latest scores.
Post-condition	1) The parent will receive the report in the mail.

3.12 Dashboards

Use Case	Reporting
Description	<ul style="list-style-type: none"> 1) User can view the performance details and progress on the dash board. 2) For local and global admin the dashboard must display the system performances.
Actors	<ul style="list-style-type: none"> 1) Global Admin 2) Local Admin 3) Kid
Preconditions	<ul style="list-style-type: none"> 1) The admin must be logged in. 2) The Kid must be logged in.
Actions	<ul style="list-style-type: none"> 1) The user profile will be dynamically updated with the latest scores. 2) Both global and local admins may define new parameters for evaluating the system
Post-conditions	<ul style="list-style-type: none"> 1) The admin can take necessary decisions about the platform's working based on the details that he see. 2) The user can view the dashboard and assess their performance.

3.13 Update profile of Admin

Use Case	Update Profile of Admin
Description	<ul style="list-style-type: none"> 1) Make the necessary changes

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

Actor	1) Admin
Precondition	1) The admin must be logged in.
Actions	1) Update personal Detail 2) Update Admin Detail 3) Switch to other Admin
Post Condition	The profile will be updated and changes will be saved.

5. Functional Requirements

A functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.

5.01 Individual User Registration

Use Case	Individual User Registration
Functional Requirements	<p>5.01.1) The system must check for the uniqueness of the username.</p> <p>5.01.2) Password constraints must be checked for validation.</p> <p>5.01.3) The platform must send a confirmation email or message to the user regarding the registration.</p> <p>5.01.4) The platform must create new user for each successful registration.</p> <p>5.01.5) The platform must create a new profile for the user with the given details.</p>

5.02 Batch User Registration

Use Case	Batch User Registration
Functional Requirements	<p>5.02.1) The system must check for the uniqueness of the username of the complete batch.</p> <p>5.02.2) Password constraints must be checked for validation.</p> <p>5.02.3) The platform must send a confirmation email or message to the local admin regarding the registration.</p> <p>5.02.4) The platform must create a batch of new users for each successful registration.</p> <p>5.02.5) The platform must create a new profile for all the users with the given details.</p>

5.03 User Login

Use Case	User Login
Functional Requirements	<p>5.03.1) The Platform must check the validation of the username and password.</p>

	<p>5.03.2) Enable the registered user to change the password if the user forgets the password.</p> <p>5.03.3) Enable a logged in user to view his/her profile.</p>
--	--

5.04 Update User Profile

Use Case	Update User Profile
Functional Requirements	<p>5.04.1) Enable the registered user to update his profile which includes various details of the user such as location, mobile number etc.,</p>

5.05 School Registration

Use Case	School Registration
Functional Requirements	<p>5.05.1) Enable the unregistered school to register to the platform.</p> <p>5.05.2) Enable the school to create a local admin.</p> <p>5.05.3) The school will be added to the list of registered schools of the platform.</p>

5.06 Selection of Content Category

Use Case	Selection of Content Category
Functional requirements	<p>5.06.1) The platform must display all the categories of the content.</p> <p>5.06.2) Enable the user to select a category.</p>

5.07 Accessing Content

Use Case	Accessing Content
Functional Requirements	<p>5.07.1) The Platform must open the content as soon as the user selects it.</p> <p>5.07.2) The platform must update the scores of the user in real time.</p>

5.08 Updating Content

Use Case	Updating Content
Functional Requirements	<p>5.08.1) Enable the global admin to add new content for various categories.</p> <p>5.08.2) Enable the global admin to remove or replace any unwanted content.</p>

5.09 Feedback

Use Case	Feedback
Functional Requirements	5.09.1) The system must provide feedback form when the user asks for it. 5.09.2) The feedback must be sent to the global admin.

5.10 Reporting

Use Case	Reporting
Functional Requirements	5.10.1) The platform must save all the actions and interactions of the user for analysis. 5.10.2) A detailed report of each child is generated and sent to the parents periodically. 5.10.3) Export the data from local admin to global admin periodically.

5.11 Dashboards

Use Case	Dash boarding
Functional Requirements	<p>5.11.1) The dashboard must be customizable according to the requirement of the admins</p> <p>5.11.2) Dashboard must provide options to add new parameters to the system to the admins.</p> <p>5.11.3) User performance dashboard must display the performance details of the user.</p> <p>5.11.4) The content dashboard must display all the available categories to the user to access.</p> <p>5.11.5) The platform must update the leader board automatically.</p>

5.12 Updating the Profile of Admin

Use Case	Update Profile of Global Admin
Functional Requirements	<p>5.12.1) Enable the global admin to update his profile which includes various details such as location, mobile number etc.</p>

6. Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

Requirements	Description
Scalability	Supports in excess of 1,000 users at a time.
Portability	Transferring the content to an offline architecture using the same framework.
Open Source	The source code will be available for continuous development.
Documentation	1. User Guide 2. API usage guidelines
Interoperability	The product should be able to add tools and the tools available on our website should be available to be added to other platforms. (LTI 1.1)
Responsive	The product will be modified according to the device size (say mobile or tablet) to be opened by the browser.

7. Learning Content

A game is a blend of two major entities, ie. A Theme and a Subject. While theme is responsible for creating the scene, a subject governs the characteristics of the objects on the scene. Together, a theme and a subject create a game.

7.1 Themes

Any content that is displayed on the platform is based on a theme, which consists of objects who behave according to the subject chosen for learning.

7.1.1 Board Game

Items	Attributes	Description
Board	1) No. of tiles	It is a static item for a level, the no. of tiles vary with each level.
Dice	1) Value to be displayed.	Generate a value on the dice randomly.
Token	1) Position	Moves the token according to dice value
Subjects	1) Addition 2) Prime Composite 3) Word to Number 4) Even and Odd	-

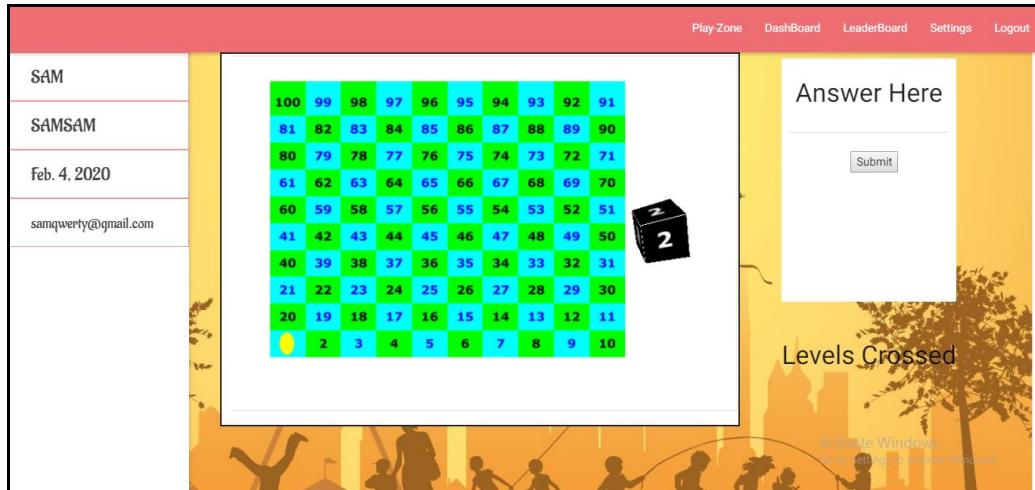


Figure: Board Game Theme

7.1.2 Candies

Items	Attributes	Description
Candies	1) Different Candy Images	Various Candies are displayed randomly
Subjects	1) Counting 2) Addition 3) Subtraction 4) Multiplication 5) Division 6) Even and Odd 7) Prime Composite	-

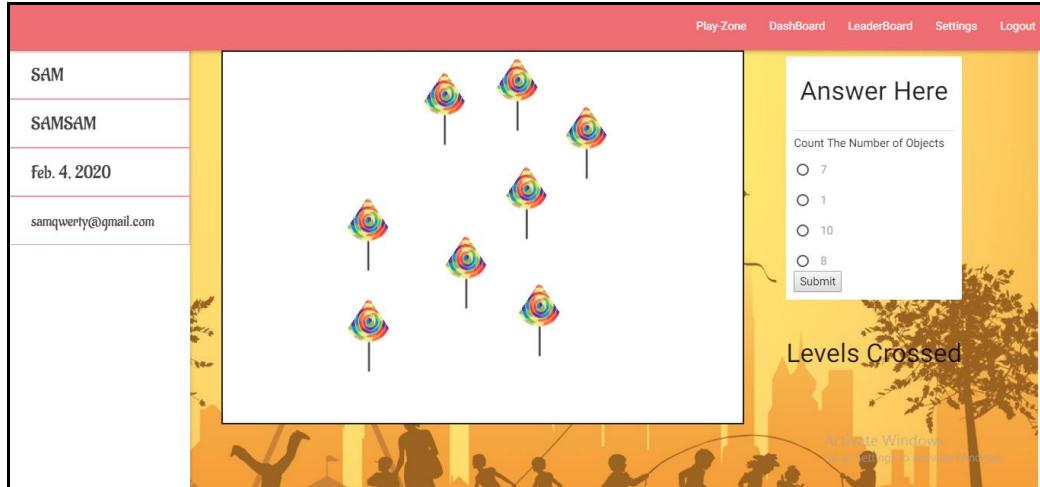


Figure: Candies Theme

7.1.3 Clock

Items	Attributes	Description
Clock	1) Clock image	Used to display the clock, static throughout the theme.
Hour and Minute Hands	1) Time to be displayed on the clock	Displays time according to various inputs
Subjects	1) Calculate the Time	-

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

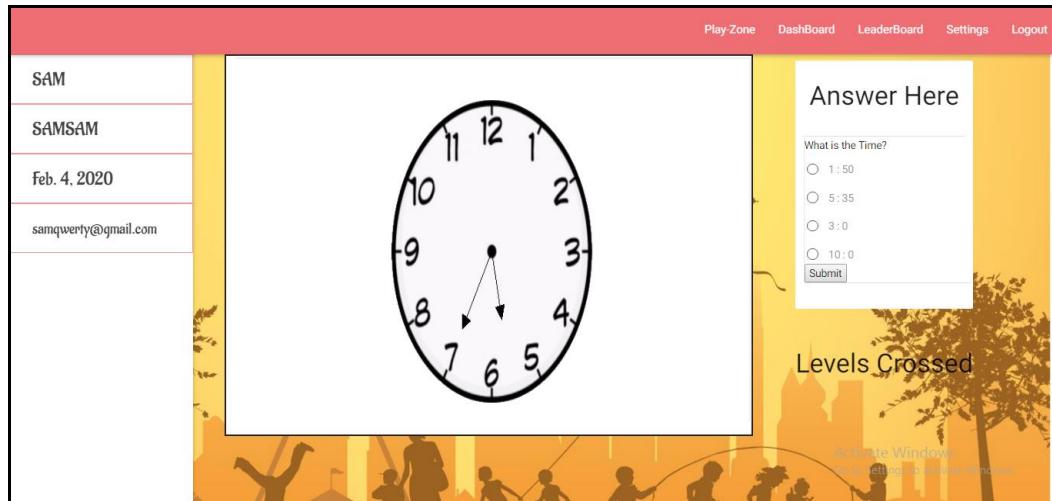


Figure: Clock Theme

7.1.4 Mine Sweeper

Items	Attributes	Description
Minesweeper Board	Clickable	Displays the numbers as required using textures
Subject	Memory	

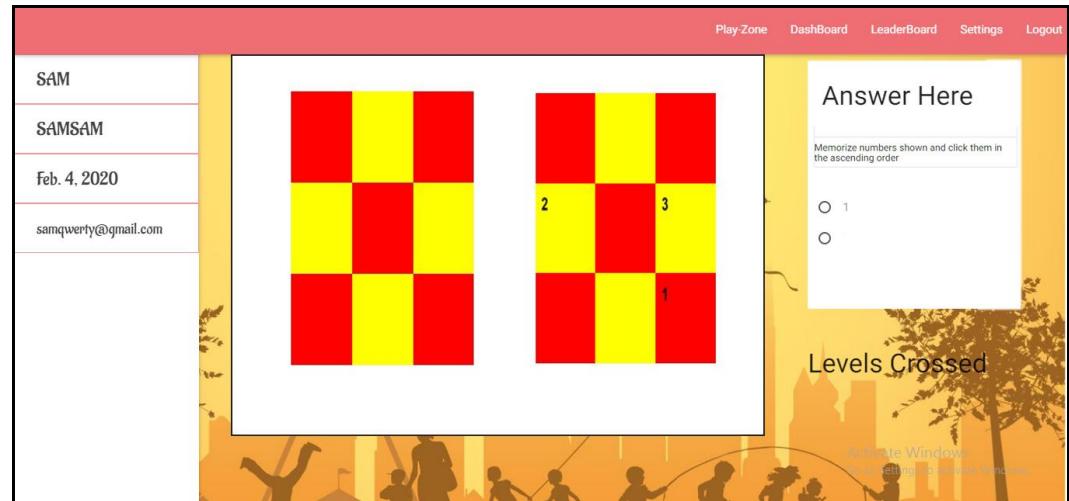


Figure: Mine Sweeper Theme

7.1.5 Number Line

Items	Attributes	Description
Number Line	1) Start range 2) End range	Used to display a number line in the range from start range to end range.
Subjects	1) Addition 2) Subtraction 3) Predecessor 4) Successor 5) Co-Primes	-

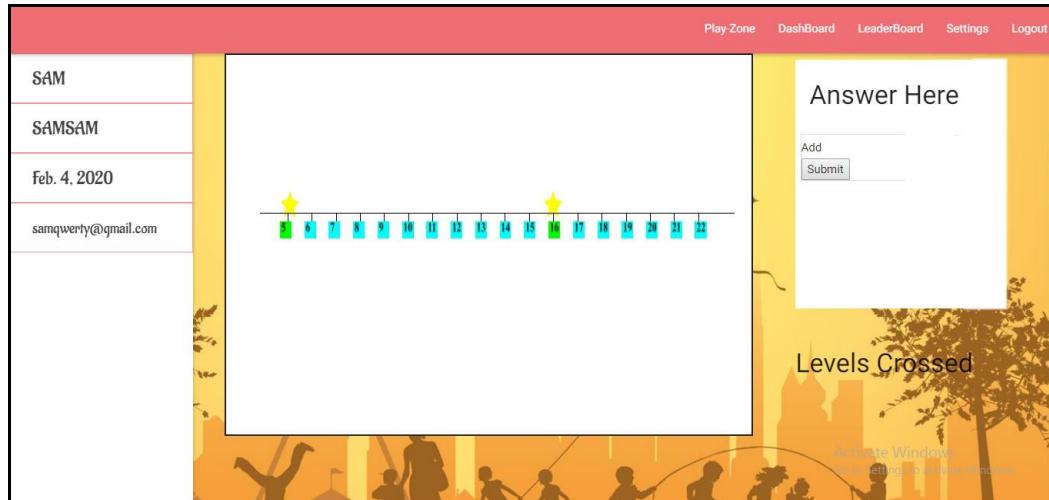


Figure: Number Line Theme

7.1.6 Numbers

Items	Attributes	Description
Text	Size, color, etc	Displays any kind of text
Subjects	1) Even and Odd 2) Prime and Composite 3) Arithmetic 4) Counting 5) LCM and GCD 6) Fractions 7) Place Value 8) Identify Number 9) Identify Name	

	10) Comparison	
	11) Coprime	
	12) Divisibility Test	
	13) Palindromes	
	14) Leap year	
	15) Days and Month	
	16) Convert case	
	17) Leading Trailing zeros	
	18) Zeros	
	19) Rhyming words	
	20) Successor and Pred.	
	21) Months	
	22) Vowels & consonants	

The screenshot shows a digital learning interface. On the left, a sidebar displays the user's name 'SAM', their role 'SAMSAM', the date 'Feb. 4, 2020', and their email 'samqwerty@gmail.com'. The main workspace contains two large green numbers, '3' and '9', positioned above a white rectangular area. To the right, a yellow sidebar titled 'Answer Here' contains a question 'Is number1 > number2' with two radio button options: 'YES' and 'NO'. A 'Submit' button is located below the radio buttons. At the bottom right, a decorative banner features the text 'Levels Crossed' over a silhouette of a city skyline and a tree. A watermark at the bottom right reads 'Activate Windows' and 'By purchasing this software you agree to the license agreement'.

Figure: Numbers Theme

7.1.7 Shapes

Items	Attributes	Description
Line	Color	Render a line
Segment	Color	Render a segment
Polygon	Color, sides	Render a polygon
Box	Color	Render a box
Cylinder	Color	Render a cylinder
Subjects	1). Shape Recognition 2). Identify Color 3). Vertex Edge Count	

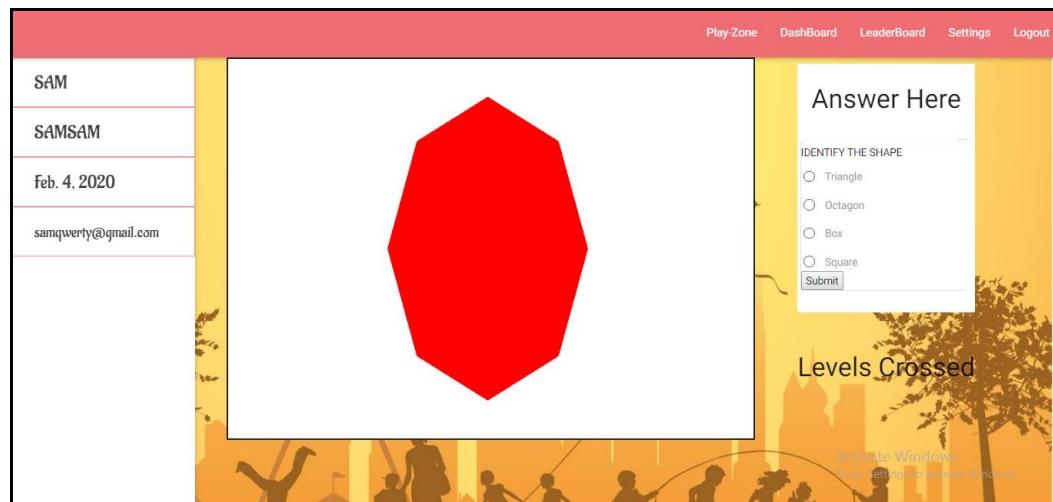


Figure: Shapes Theme

8. System Design

8.1 Brief Description

Module Name	Description
User Interface	For client side interaction.
User Profile (Dynamic, Static and Admin Profile, LDAP)	For storing the dynamic details (scores, recent activity etc.)
Game Engine	Executes (starts) the Learning Content
Themes Repository	For storing the themes related data.
Subject Repository	For storing the subjects related data.
Feedback Repository	For storing the feedbacks.
Reporting Engine	For generating and sending the reports
Analytics Engine	For fetching statistics related to the website.

8.2 Module Interaction Diagram

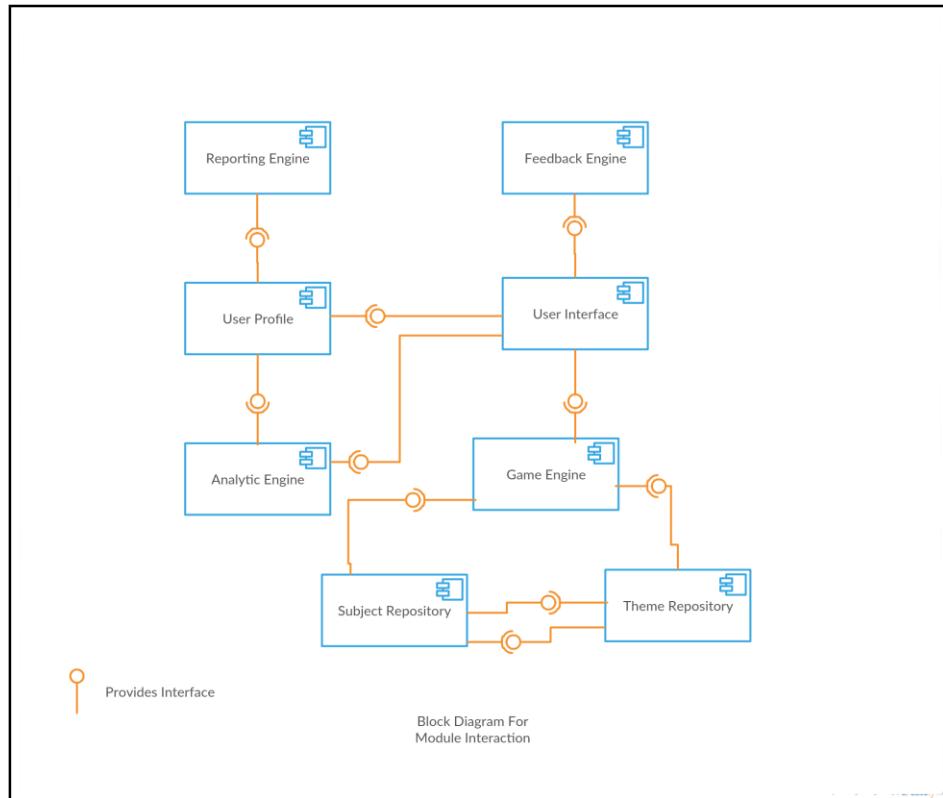


Figure: Block Diagram for Module Interaction

8.1 User Interface

8.1.1 Description

This module will handle all the client side interaction for the platform.

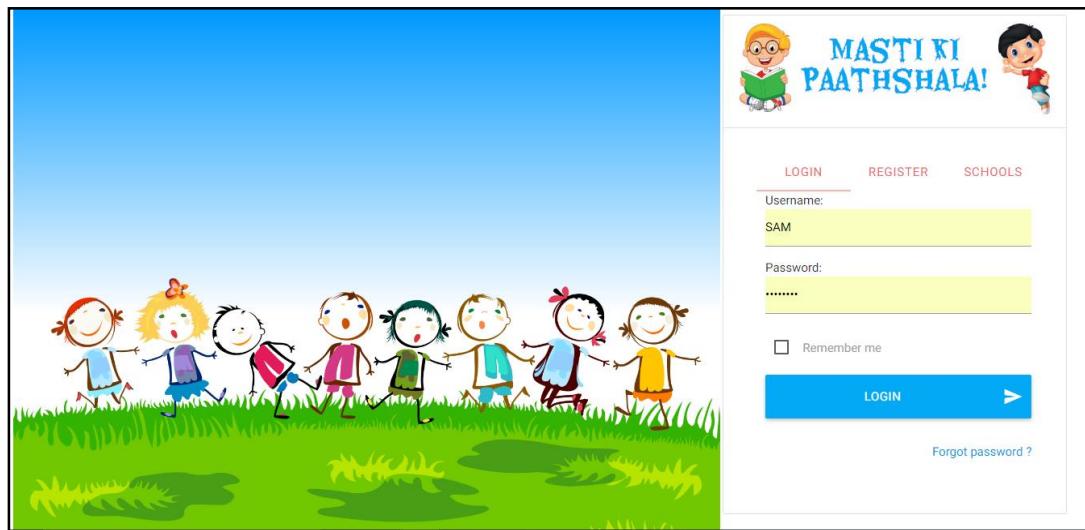


Figure: Home Page

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

The screenshot shows the user profile for 'SAM' (SAMSAM, Feb. 4. 2020, samqwerty@gmail.com). The interface is divided into three main sections: MATHS, BRAIN, and a central content area.

- MATHS:**
 - addition:** Level : 0
 - NUMBERS:** Level : 0
 - numLineSuc:** Level : 0
 - NUMBERLINE:** Level : 0
- BRAIN:**
 - IdentifyName:** Level : 0
 - NUMBERS:** Level : 0

Figure: Learning Content

The screenshot shows the user profile for 'SAM' (SAMSAM, Feb. 4. 2020, samqwerty@gmail.com). The interface features a central game area with colorful lollipops and a question on the right side.

Answer Here:

Count The Number Of Objects

7
 1
 10
 8

Submit

Levels Crossed

Activate Windows
Go to Settings to activate Windows

Figure: Game Play

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

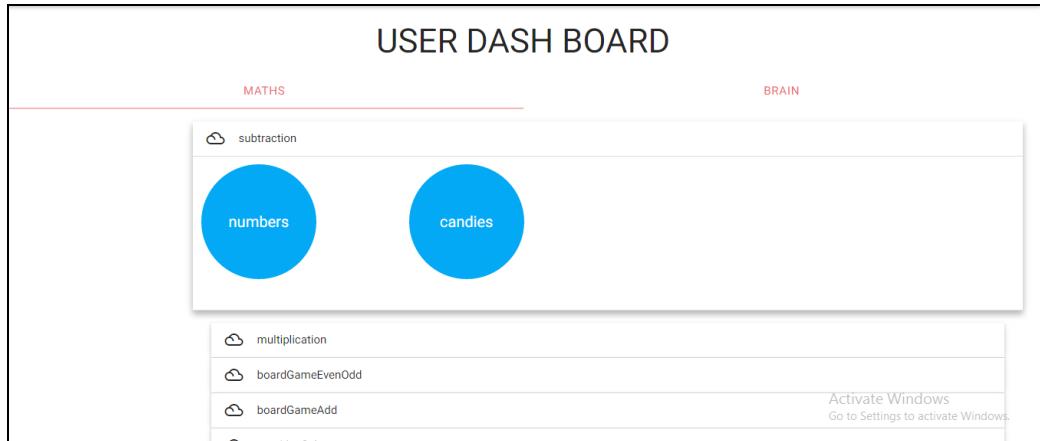


Figure: User Dashboard Collapsed

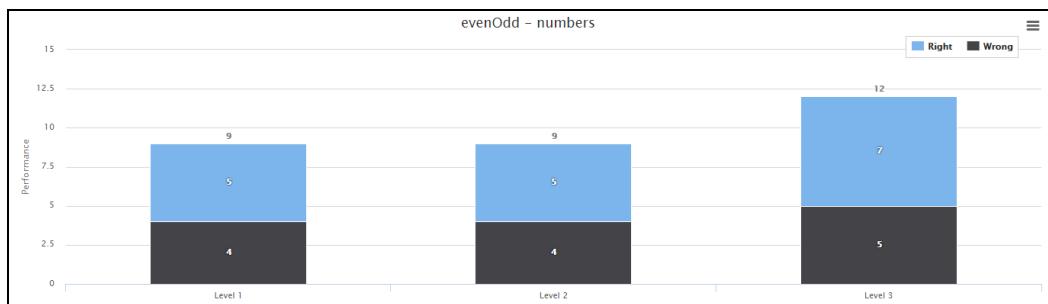


Figure: User Dashboard Expanded

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

The screenshot shows a red header bar with navigation links: Play-Zone, DashBoard, LeaderBoard, Settings, and Logout. On the left, there's a sidebar with user information: SAM, SAMSAM, Feb. 4, 2020, and samqwerty@gmail.com. The main area is titled "LEADERBOARD" and displays a table of users:

Rank	User Name	School Name
1	SAM	QWERTY
2	aditya	klikop
3	SAMS	IITBOMBAY

Below the table is a search bar labeled "Search By Name/School" and a green "SUBMIT" button. At the bottom right, there's a message: "Activate Windows Go to Settings to activate Windows."

Figure: Leader board

The screenshot shows a dark blue header bar with "Admin Panel" on the left and "WELCOME, MAS. VIEW SITE / CHANGE PASSWORD / LOG OUT" on the right. The main area is titled "Site administration". It has two sections: "APP" and "AUTHENTICATION AND AUTHORIZATION".

APP

Compatibility Matrices (Brain)	+ Add	Change
Compatibility Matrices (Math)	+ Add	Change
Subjects	+ Add	Change
Themes	+ Add	Change

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

Recent Actions

My Actions

- SAM User

Figure: Admin Panel

8.1.2 Components

Components	Description
Home Page	This is the first page which contains register/login tabs, About, and general website

	related information links (Copyright, sitemap etc.)
Registration Page	Used for registering a user, opens when the user clicks on the register tab on the home page
Login Page	Once registered, a user can login through his/her User ID and Password.
User Profile	After Successful login, this page appears. It displays various subjects, the related themes and recent activity of user. Contains basic information about the user (Name, Age, School etc.) in a sidebar navigation.
Learning Content	The actual learning content page, where user will play/interact with the learning content. It is further explained in the Theme and Subject modules
User Leaderboard	Contains information about the users and their points in various learning contents.

8.1.3 Technologies Used

Technologies	Description
Google Material Design Library	Used for designing the front end for the platform. It is a lightweight

	and open source CSS/JS library with very attractive UI customization options.
--	---

8.1.4 Requirements Met

Requirements
5.03.1
5.04.1
5.06.1, 5.06.2
5.07.1, 5.07.2
5.08.1, 5.08.2
5.11.3, 5.11.4, 5.11.5, 5.11.6
5.12.1

8.2 User Profile Repository

8.2.1 Description

User Profile Repository module consist of details regarding all the users including admins and information about user groups.

8.2.2 Components

Components	Description
LDAP Authentication Database	LDAP is a central place to store usernames and passwords. This will allow the platform to authenticate the users.

Static Use Profile Database	This database stores the general information about the user
Dynamic User Profile Database	This database stores that information about the user which changes frequently, like scores, recent activity etc.
Admin Profile Database	This database stores general information about the admin.
School Database	This database stores information about a special type of group ie. School.

8.2.3 Technologies Used

Technologies	Description
MySQL	It is the most widely used open source client-server model RDBMS
Django	It is the one of the most used backend framework. It has an excellent documentation, very simple object-relation mapping, clear separation of template / controller, Django-admin support and pluggable apps.

8.2.4 Requirements Met

Requirements

5.01.1,5.01.2,5.01.3,5.01.5,5.01.5
5.02.1,5.02.2,5.02.3,5.02.4,5.02.5
5.03.1,5.03.2,5.03.3
5.04.1
5.05.1,5.05.1,5.05.3
5.06.2
5.07.1,5.07.2
5.11.1,5.11.2,5.11.4,5.11.5,5.11.6
5.12.1
5.13.1

8.2.5 Schema Description

8.2.5.1 Static User Profile

Attributes	Description
Username	Used as a primary key for unique identification of the kid
Name	Name of the Student
School/Day Care Centre	The School/Day Care Centre the user belongs to
Date of Birth	Date of Birth to obtain the age of the kid/
Parent's email	To send the performance reports of the kid to the parent via email.
Parent's contact number	To send the performance reports of the kid to the parent via SMS.

Address	For Analytical purposes
City	For Analytical purposes
State	For Analytical purposes

8.2.5.2 Dynamic User Profile

Attributes	Description
Username	Used as a primary key for unique identification of the kid
Subject	The subject used
Theme	The theme used
Level	Level the kid is in.
Score	Kids score in the game
Last Played	Timestamp to keep track of recent activity

8.2.5.3 Admin Profile

Attributes	Description
Username	Used as a primary key for unique identification Of the admin
Name	Name of the Admin
Date of Birth	To obtain the age of the user
Address	For Analytical purposes
City	For Analytical purposes
State	For Analytical purposes
Email	Contact Information

Contact Number	Contact Information
----------------	---------------------

8.2.5.4 School Database

Attributes	Description
School ID	Used as a primary key for unique identification Of the school
Group ID	Used as a primary key for unique identification Of the school with respect to group.
School Name	Name of the School
School Registration Number	For the Validation if the school
School Address	For Analytical purpose.
Local Admin ID	For the unique identification if the Local Admin.

8.2.6 ER Diagram

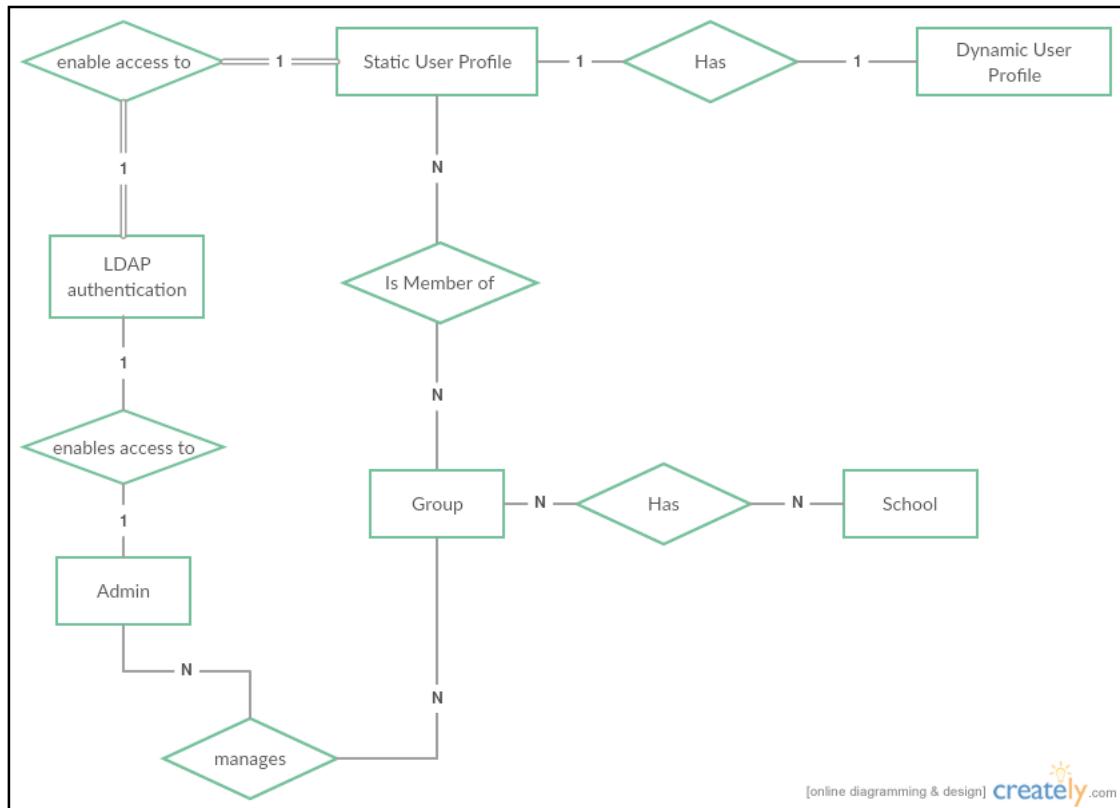


Figure: ER Diagram for User Profile

8.3 Feedback Engine

8.3.1 Description

This module will provide a medium for feedback about the system for any user. The user could be one who has registered on the platform or an anonymous person. The global

administrator will be able to view the feedbacks from the database.

8.3.2 Components

Components	Description
Feedback Form	Contains text field for the user to write the feedback and submit it to the admin
Feedback Database	A Database which contains all the feedback submitted which can be accessed by the admin

8.3.3 Technologies Used

Technologies	Description
Google Material Design Library	Used for designing the front end for the platform. It is a lightweight and open source CSS/JS library with very attractive UI customization options.
MySQL	It is the most widely used open source client-server model RDBMS

8.3.4 Requirements Met

Requirements
5.09.1, 5.09.2

8.3.5 Schema Description

8.3.5.1 Feedback Database

Attributes	Description
Feedback Id	Automatically generated and auto incremented by the database
Username	For storing the username (null if anonymous)
Comment Text	The feedback field which would be filled by the user.
Date	Auto generated by the system.

8.4 Reporting Engine

8.4.1 Description

This module will fetch the user performance details from the dynamic user profile. A detailed report will be generated for each user and will be sent periodically to the parent/guardian by sending an email and a text message. The module will periodically send the details of the group to the global admin, if he wishes to share the details.

8.4.2 Components

Components	Description
Report	Various details about the user will be sent in the report like Name of the user, Subject completed

	recently, scores, timestamp of the last played game.
--	--

8.4.3 Technologies Used

Technologies Used	Description
EMail API	IMAP/POP3 mail service for sending/receiving mails through the website

8.4.4 Requirements Met

Requirements
5.10.2, 5.10.3

8.5 Analytics Engine

8.5.1 Description

This module will provide a medium for the admin to get live statistics about the platform.

8.5.2 Components

Components	Description
Page Views	A metric defined as the total number of page views.
Unique Visitors	The <i>Users</i> and <i>Active Users</i> , and unique visitors metrics show how

	many users viewed or interacted with the platform
Average Session Duration	How much time the user spent on the platform in a single session
OS Details	Details about the OS used to access the platform

8.5.3 Technologies Used

Technologies Used	Description
Django Admin	Django Admin is an in-built admin panel in any Django app. It can be used to manage the app and see the stats for the app as an Admin

8.5.4 Requirements Met

Requirements
5.11.1,5.11.2,5.11.3

8.6 Game Engine

8.6.1 Description

This module will provide a medium for rendering the theme and subject together to make the learning content.

8.6.2 Components

Components	Description
------------	-------------

Three.JS Library	Used for rendering all the graphics
File Reader	This component will read the Theme and Subject JSON files.
Main Function	This component will initialize the level, scores etc. for the user for that particular level, and calls the execute function which is in the API Library to start the game execution
Execute Function	It is the main function in the API Library, it evaluates the theme, subject functions and makes logic out of it.
ExecuteForm Function	Evaluates the form json file and renders the user input form on the screen
Decision Engine	Based on the user input, the assessment is done, either the positive count is incremented or the negative one. Both will be sent in the report to the guardian for evaluation.

8.6.3 Technologies Used

Technologies Used	Description
Three.js	Library used to build the content

8.6.4 Requirements Met

Requirements
5.07.1,5.07.2
5.08.1,5.08.2

8.6.5 Algorithm

Following algorithm was developed for the game engine:

```

Import Three.js
Import CentralAPILibrary
Subject selected – Subject file loaded
Theme selected – Theme file loaded
Function gameEngine ()
{
    Initialize Variables
    Level = user.dynamicProfile.subject.theme.Level
    Score = user.dynamicProfile.subject.theme.Score
    Attempt = user.dynamicProfile.subject.theme.Attempt
    s = parse (subject)
    t = parse (theme)
    Render SceneL (Level Begins)
    Remove the SceneL
  
```

```

Init(rendererCount)

function init (count)

{
    Render the canvas according to count
    Render the scenes, cameras, renderers
}

apilib.execute(level,s,t,attempt)

}

function execute(level,s,t,attempt)

{
    If(level == more than actually present)
        Return true;

    for(var k = 0; k< executeCount; k++)
    {
        ---Execute all subject functions;
        Puts what has to be rendered on the screen in the variable
        'correctAns'
        Puts the unique operation in the variable 'operation'
        ---Execute all theme functions
        Actually render the content according to the theme
        correctAnsArray.push(correctAns);//stores all 'correctAns'
    }
    executeForm (level,correctAnsArray,operation,attempt)
}

```

```
function executeForm(level,correctAnsArray,operation,attempt)
{
    array = apilib.calculate (correctAnsArray,level,operation)
    //calculate puts the correct answers in the array to be compared with
    the user input
    form = parse(form)
    Create the input form as per the type of input specified in JSON file
    Create the submit button
    On button click, according to the type, action taken and user input
    evaluated comparing it with the correct answers calculated in the
    method 'calculate'
    apilib.execute(level,s,t,attempt) //rendering the new question
}
}
```

Algorithm: Game Engine

8.6.6 Flow Diagram for Renderer

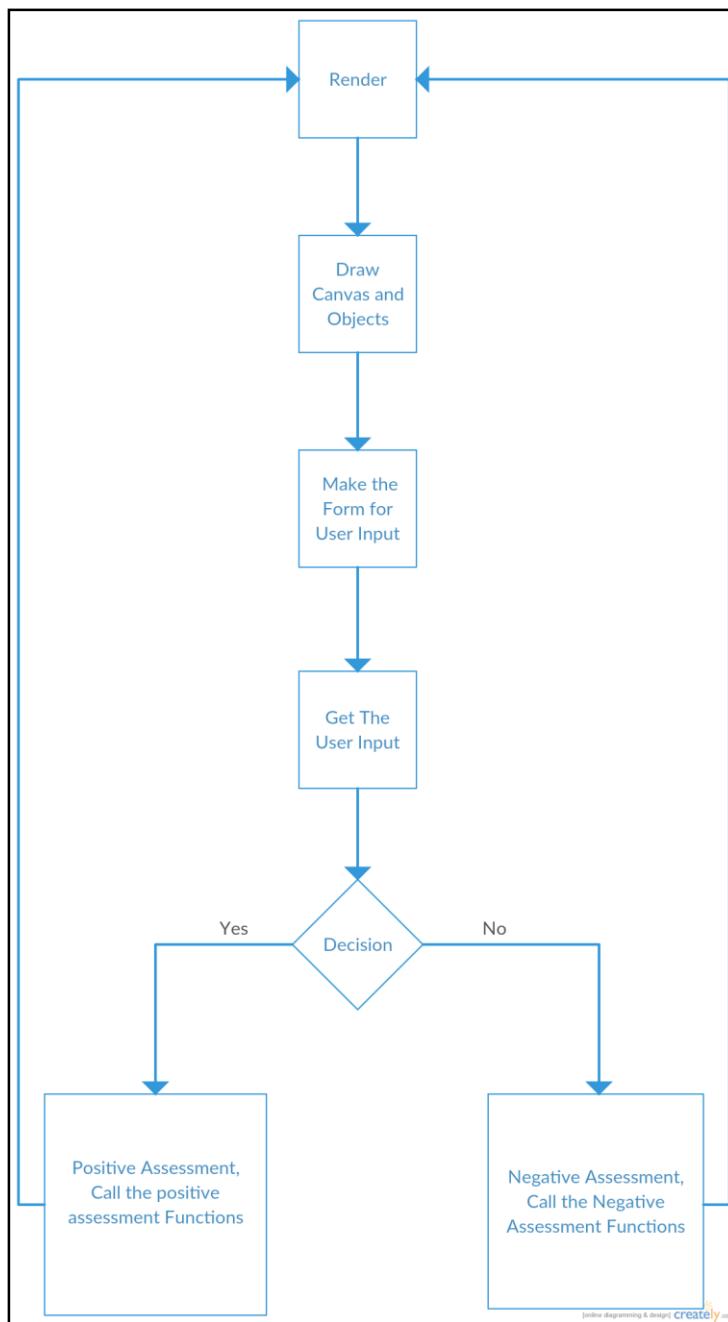


Figure: Flow Diagram for Renderer

8.7 Theme Repository

8.7.1 Description

This module will contains all the theme related files which contains the objects of the theme and the functions which are to be called during state transitions.

```

1  //JSON File Format for a Theme Creation
2  //Define everything inside the theme variable
3  //Theme = [functions,obj1,obj2,...] as an array of objects
4
5  theme = '
6  [
7      //The Main theme functions are defined here
8      "functions":
9      [
10         {
11             "returnValue":"",
12             "functionName":"",
13             "params": //The parameters of this function
14             [
15                 {
16                     "key":"",
17                     "data": "" //the actual value of the parameter
18                 },
19                 {
20                     "key":"",
21                     "data": "" //the actual value of the parameter
22                 }
23             ],
24             "returnValue":"",
25             "functionName":"",
26             "params":
27             [
28                 {
29                     "key":"",
30                     "data": ""
31                 }
32             ]
33         },
34         //After functions, a theme may have various objects to be rendered, define them here
35         //The name should be identical to that of the JSON variable of this object
36         //see dummy_object.json to get an idea
37         {
38             "name": "dummy_object"
39         }
40     ]
'

```

Figure: Theme JSON File Format

```

1 //JSON File Format for a Theme Creation
2 //Define everything inside the dummy_object variable //You can change the name of this object
3 //Make sure to include it in your theme.json file by this name
4 //dummy_object = [functions] as object
5
6 dummy_object = '
7 [
8     //The Main object functions are defined here
9     "functions" :
10    [
11        {
12            "functionName" : "", //name of the function that is to be called in the APILIB
13            "params": //The parameters of this functions
14            [
15                {
16                    "key":"", //key is the name of the param (not used as such)
17                    "data":"" //the actual value of the parameter
18                } //Define more params here
19            ]
20        } //Define more functions here
21    ]
22].

```

Figure: Object JSON File Format

8.7.2 Components

Components	Description
Theme JSON Files	Contains brief description of the objects which link to the actual object JSON Files and also sequence of function calls to be called by the game engine.
Object JSON File	It contain the description each and every object and methods.

8.7.3 Technologies Used

Technologies Used	Description
JavaScript	Language to store the theme details in JSON.

8.7.4 Requirements Met

Requirements
5.07.1, 5.07.2
5.08.1, 5.08.2

8.8 Subject Repository

8.8.1 Description

This module will contains all the subject related files which contains the attributes which store the information regarding a particular level and the functions which are to be called during state transitions.

```

1 //JSON File Format for a Subject Creation
2 //Define everything inside the subject variable
3 //Theme = [[functions,assessmentCount,],executeCount,rendererCount] as an array of objects
4
5 subject = [
6   {
7     //The Main theme functions are defined here for a level
8     "functions": [
9       {
10         "returnValue": "", //what the function returns, either a value or a variable name already defined in APILIB
11         "functionName": "", //name of the function that is to be called in the APILIB
12         "params": //The parameters of this function
13         [
14           {
15             "key": "", //key is the name of the param (not used as such)
16             "data": //the actual value of the parameter
17           },
18           {
19             "key": "",
20             "data": ""
21           } //Add more here
22         ],
23       },
24       //this is a mandatory function, it is to be defined in every subject file
25       "returnValue": "operation",
26       "functionName": "operate",
27       "params": [
28         {
29           "key": "perform",
30           "data": //Add the number according to the case number in operate function of APILIB
31         }
32       ],
33       "assessmentCount": 3 //The number of correct answers, the user has to give, for crossing this level
34     }, //add more levels here
35   },
36   "executeCount": 1 //number of times the functions need to be executed to obtain the values
37 },
38 {
39   "rendererCount": 1 //number of renderers required by the game (generally it is equal to the execute count
40   //If it is greater than the execute count , you can add different things in the other renderers
41 }]
]

```

Figure: Subject JSON File Format

```

1  //JSON File Format for a Form Creation
2  //Define everything inside the form_a variable
3  //form_a = [{questions,types,dynamic,functions,labels}] as objects array
4  form_a =
5  [
6      {
7          "questions": //Define all your questions here as key-value pair
8          {
9              "key": "",
10             }///add more here
11         },
12         "types": //what type of options are required
13         [
14             {
15                 "key": "radio", //radio = radioButton
16                 "label": text view
17                 //"noSubmit" = no button required
18                 //"1" = nothing but plain submit button required
19                 "count": //the number of entities required
20             },
21             "dynamic": //If functions are required or not
22             [
23                 "key": //either true or false //if true, define functions below, else define the labels
24             ],
25             "functions": //Name of the functions to create dynamic options, for each question
26             [
27                 //for q1
28                 "functionName": "", //Name of the function
29                 "params": //the parameters it requires
30                 [
31                     {
32                         "key": "", //key is the name of the param (not used as such)
33                         "data": //the actual value of the parameter
34                     },
35                     {
36                         "key": "",
37                         "data": //
38                     }
39                 ]///Define more according to number of questions
40             ],
41             "labels": //labels are required if you wish to manually add the options
42             [ //Define for q1
43                 {
44                     "params": [
45                         {
46                             "key": "" //Define the name here
47                         }
48                     ]///add more here
49                 }
50             ]
51         ]
52     ]
53 ]

```

Figure: Form JSON File Format

8.8.2 Components

Components	Description
JSON Files	Each and every details related to subject such as attributes describing a particular level like range, sequence of function calls will be stored in the JSON Files which is to be read by the game engine.
Form JSON File	This file contains the details about the user input form, and functions to create the options. Each subject has its own unique form json file.

8.8.3 Technologies Used

Technologies Used	Description
JavaScript	Language to store the subject details in JSON.

8.8.4 Requirements Met

Requirements
5.07.1,5.07.2
5.08.1,5.08.2

9. Testing

9.1 Description

9.1.1 Objective

Objective of Test plan is to define the various Testing strategies and testing tools used for complete Testing life cycle of this project.

9.1.2 Scope

Testing mainly targets the GUI testing and validating data in report output as per Requirements Specifications provided by Client.

The platform has been fully tested, all possible test cases have been verified and test cases are mentioned below.

9.2 Test Design Description

9.2.1 Features Tested

Requirement ID	Feature	Tested
5.01	<u>User Registration</u>	Yes
5.02	<u>Group Registration</u>	Yes
5.03	<u>User Login</u>	Yes
5.04	<u>Update User Profile</u>	Yes
5.05.1	<u>Add or Remove Member from Group</u>	No
5.05.2	<u>Update Group</u>	No
5.06	<u>Selection of Content Category</u>	Yes

5.07	<u>Accessing Content</u>	Yes
5.08	<u>Updating Content</u>	Yes
5.09	<u>Feedback</u>	No
5.10	<u>Reporting</u>	No
5.11	Dashboard Creation	Yes
5.12	<u>Updating Global Admin Profile</u>	Yes

9.3 Testing Strategy

9.3.1 Description

A test strategy is an outline that describes the testing approach of the software development cycle. This includes the testing objective, methods of testing new functions.

9.3.2 Types

9.3.2.1 Black-Box

It is sometimes called behavioral testing or Partition testing. This kind of testing focuses on the functional requirements of the software. It enables one to derive sets of input conditions that will fully exercise all functional requirements for a program.

9.3.2.2 GUI Testing

GUI testing will include testing the UI part of report. It covers users Report format, look and feel, error messages, spelling mistakes, GUI guideline violations.

9.3.3 Bug Life Cycle

Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime.

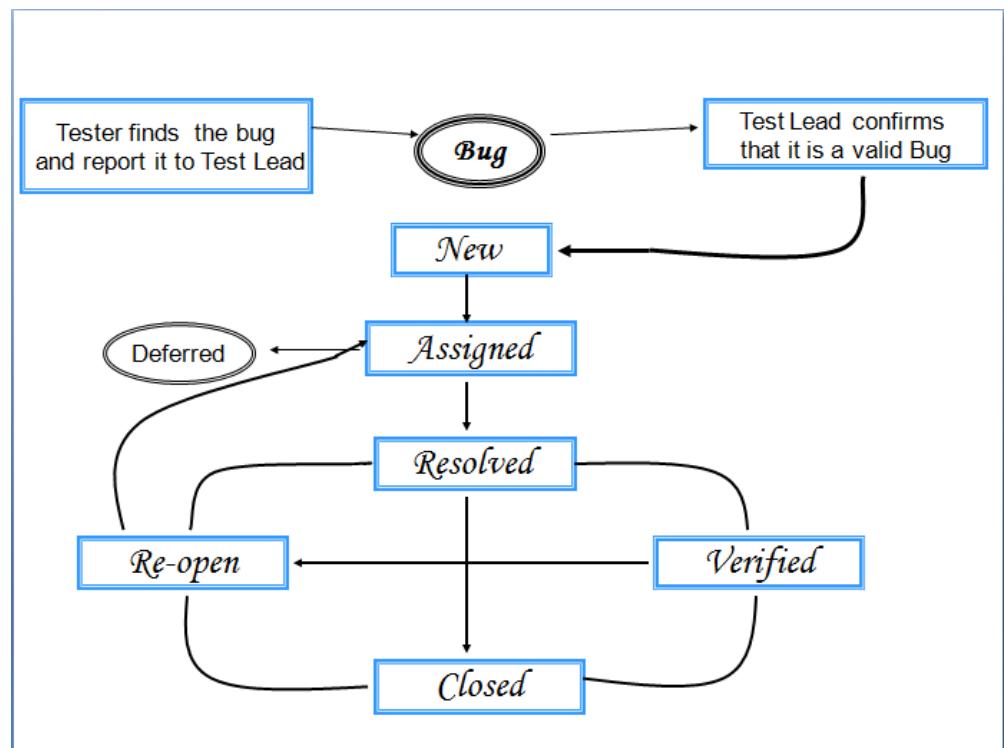


Figure: Bug Life Cycle

9.4 Testing Identification

9.4.1 User Registration

Parameter Name	Description
Name	It contains the kid's name.
Date of Birth	It contains the Date of Birth.
Residential Address	It contains the residential address of kid.

Parent Email-Id	It contains parent email-id.
Parent Contact Info	It contains parent contact information.
Gender	It contains the gender of kid.
Username	It contains the appropriate username.
Password	It contains the appropriate password.
Confirm Password	It contains the same password as in the password field.
School Name	It contains the kid's school name.

9.4.2 Group Registration

Parameter Name	Description
Group Name	It contains the name of the group.
Group Admin Name	It contains the name of the group admin.
Group Admin Email-Id	It contains the email-id of group admin.
Group Admin Contact Info	It contains group admin contact information.

9.4.3 User Login

Parameter Name	Description
Username	It contains the username.
Password	It contains the password.

9.4.4 Update User Profile

Parameter Name	Description
Name	It contains the kid's name.
Date of Birth	It contains the Date of Birth.

Residential Address	It contains the residential address of kid.
Parent Email-Id	It contains parent email-id.
Parent Contact Info	It contains parent contact information.
Gender	It contains the gender of kid.
Username	It contains the appropriate username.
Password	It contains the appropriate password.
Confirm Password	It contains the same password as in the password field.
School Name	It contains the kid's school name.

9.4.5 Update Group

Parameter Name	Description
Group Name	It contains the name of the group.
Group Admin Name	It contains the name of the group admin.
Group Admin Email-Id	It contains the email-id of group admin.
Group Admin Contact Info	It contains group admin contact information.

9.4.6 Add/Remove Members to/from a Group

Parameter Name	Description
Username	It contains the username.

9.4.7 Dynamic Dashboard

Parameter Name	Description
Information Field	The field based on which the admin wants to generate the dashboard.
Name of New Parameter	The new parameter that is to be added into the system.
Username	It contains the username of the kid.
Category Name	It contains the required category that is to be accessed.

9.4.8 Dynamic User Profile

Parameter Name	Description
Username	It contains the username.

9.4.9 Manual Dashboard Creation by Admin

Parameter Name	Description
X_Parameter	It contains the parameter along the X-axis.
Y_Parameter	It contains the parameter along the Y-axis.

9.4.10 Reporting

Parameter Name	Description
Username	It contains the kid's username.
Parent Email-Id	It contains parent email-id.
Parent Contact Info	It contains parent contact information.

9.4.11 Feedback

Parameter Name	Description
Name	It contains the sender's name.
Email-Id	It contains sender's email-id.
Contact Info	It contains senders contact information.

9.4.12 Themes (Board Game Theme)

Parameter Name	Description
No of Tiles	Describes number of tiles required.
Dice	It contains the number to be displayed on the dice.
Token Position	It contains the position of the token.

9.4.13 Subjects

Parameter Name	Description
Child's Response	It contains the response of the child.

9.5 Testing Results

Requirement ID	Feature	TESTING NATURE	TEST RESULT FILE
5.01	<u>User Registration</u>	AUTOMATED	userregres.xlsx
5.02	<u>Group Registration</u>	MANUAL	-
5.03	<u>User Login</u>	AUTOMATED	userlogres.xlsx
5.04	<u>Update User Profile</u>	AUTOMATED	userprores.xlsx
5.05.1	<u>Add or Remove</u>	-	-

	<u>Member from Group</u>		
5.05.2	<u>Update Group</u>	-	-
5.06	<u>Selection of Content Category</u>	AUTOMATED	gameres.xlsx
5.07	<u>Accessing Content</u>	MANUAL	
5.08	<u>Updating Content</u>	MANUAL	
5.09	<u>Feedback</u>	-	-
5.10	<u>Reporting</u>	-	-
5.11	<u>Dashboard Creation and Leaderboard</u>	AUTOMATED	dashleadres.xlsx
5.12	<u>Updating Global Admin Profile</u>	MANUAL	-

9.6 Manual Testing Results

9.6.1 Game Testing

9.6.1.1 Even and Odd – Board Game

Bug ID 1: The options presented on the screen were wrong

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

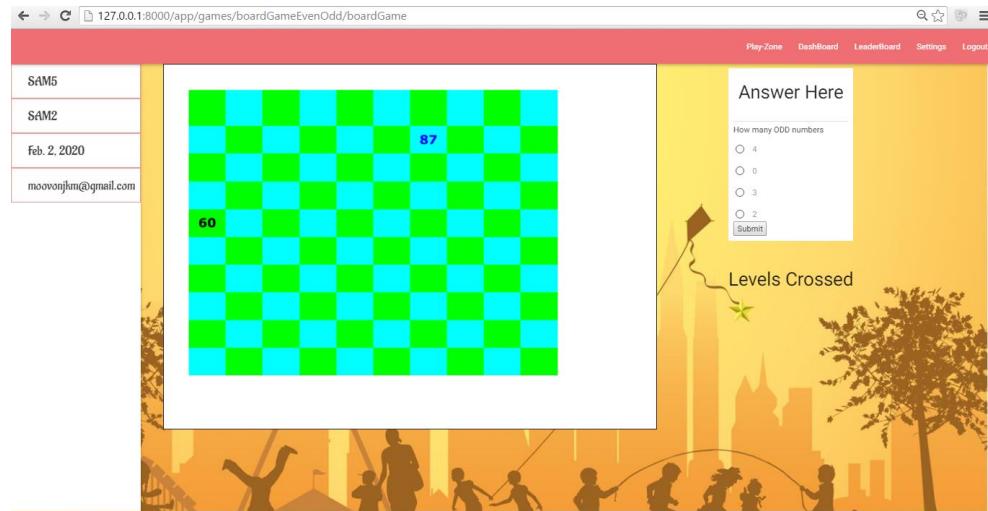


Figure: Bug ID 1

9.6.1.2 Prime and Composite – Numbers

Bug ID 2: Number 1 displayed on the screen, which is neither prime nor composite.

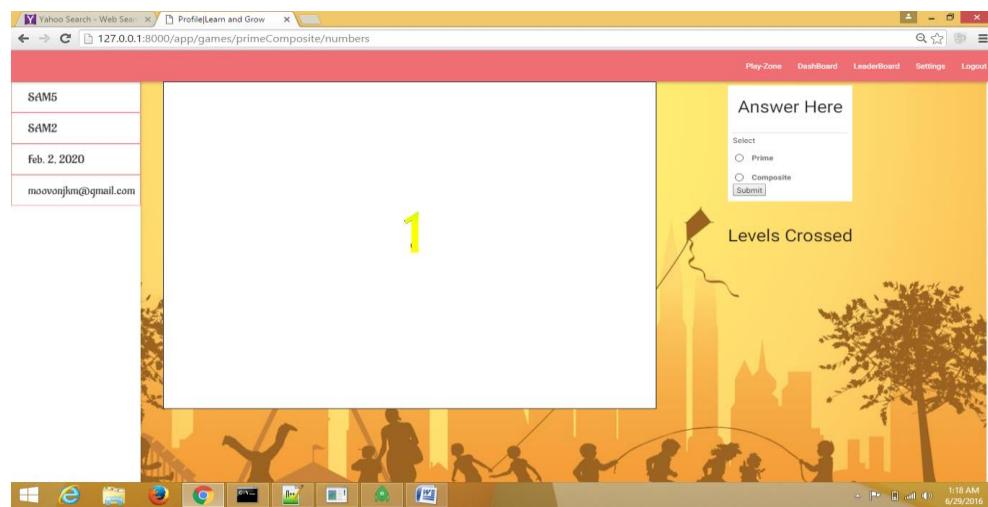


Figure: Bug ID 2

9.6.1.3 Prime and Composite – Board Game

Bug ID 3: Some prime numbers were not clickable.

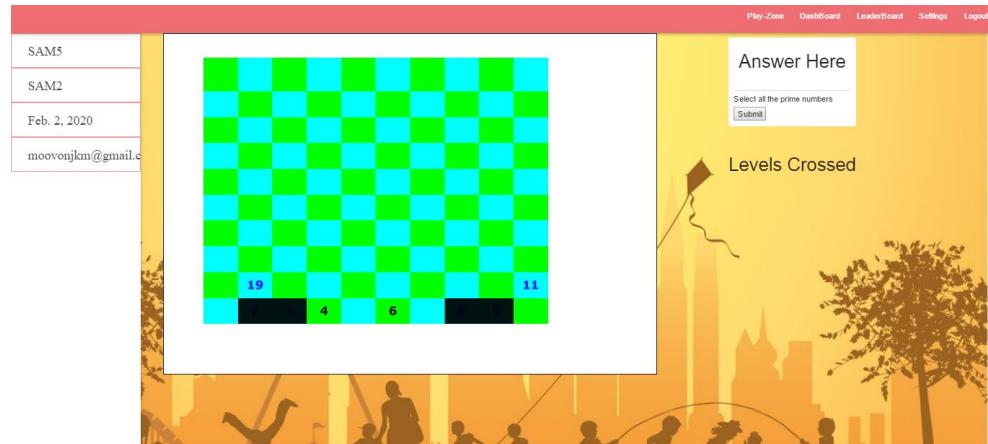


Figure: Bug ID 3

9.6.1.4 Fraction Arithmetic – Numbers

Bug ID 4: Other than the correct answer, other answers were not displayed as fraction.



Figure: Bug ID 4

9.6.1.5 Addition – Board Game

Bug ID 5: Issue with Token movement



Figure: Bug ID 5

9.6.1.6 Days and Months – Numbers

Bug ID 6: Wrong Spellings

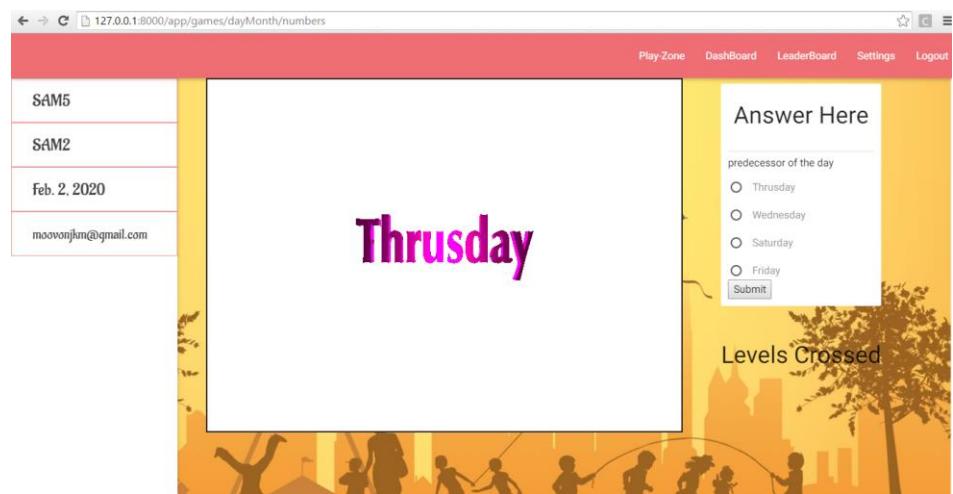


Figure: Bug ID 6

9.6.1.7 Blank Screen on Game Completion

Bug ID 7: Blank Screen on Game Completion

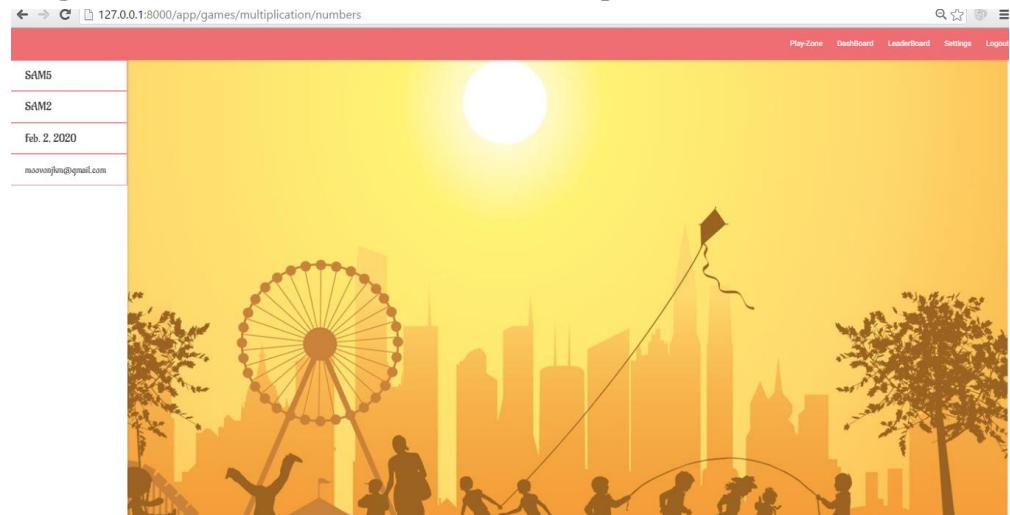


Figure: Bug ID 7

9.6.2 Testing of Platform

9.6.2.1 Leaderboard

Bug ID 8: Leaderboard was not accessible.

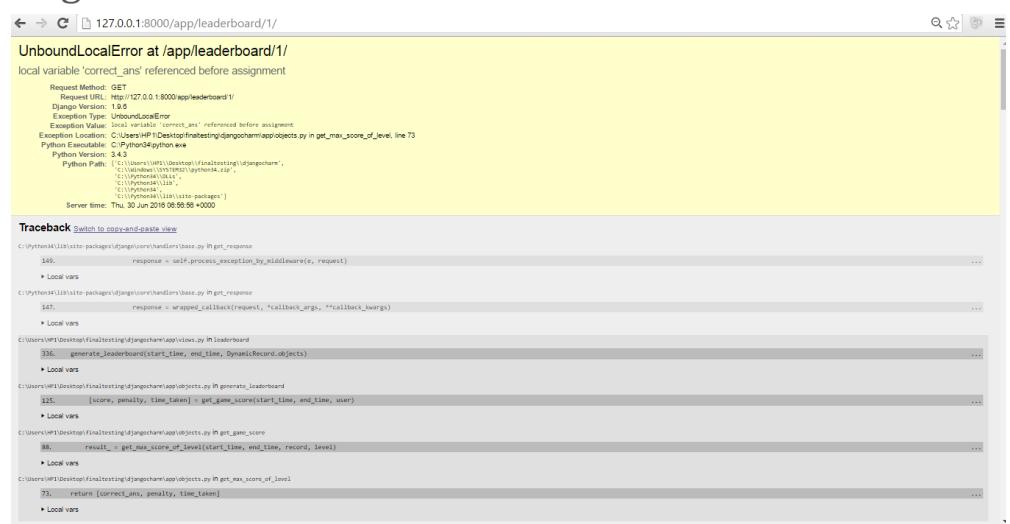


Figure: Bug ID 8

9.6.2.2 Dashboard

Bug ID 9: Dashboard was not accessible.

```

UnboundLocalError at /app/dashboard
local variable 'correct_ans' referenced before assignment

Request Method: GET
Request URL: http://127.0.0.1:8000/app/dashboard
Django Version: 1.9.8
Exception Type: UnboundLocalError
Exception Value: local variable 'correct_ans' referenced before assignment
Exception Location: C:\Users\HP\Desktop\maths\project\djangochar\app\objects.py in get_max_score_of_level, line 59
Python Executable: C:\Python34\python.exe
Python Version: 3.4.3
Python Path: ['C:\\Users\\HP\\Desktop\\maths\\project\\djangochar\\', 'C:\\\\Windows\\\\System32\\\\python34\\\\', 'C:\\\\Windows\\\\System32\\\\', 'C:\\\\Windows\\\\System32\\\\\\site\\packages\\']
Server Time: Thu, 30 Jun 2016 08:11:29 +0000

Traceback (most recent call last):
  File "C:\Python34\lib\site-packages\django\core\handlers\base.py" in get_response
    149.         response = self.process_exception_by_middleware(e, request)
  File "C:\Python34\lib\site-packages\django\core\handlers\base.py" in get_response
    147.         response = wrapped_callback(request, *callback_args, **callback_kwargs)
  File "C:\Python34\lib\site-packages\django\utils\decorators.py" in wrapped_callback
    14.     return wrapped(*args, **kwargs)
  File "C:\Users\HP\Desktop\maths\project\djangochar\app\views.py" in dashboard
    59.     level = Level.objects.get_max_score_of_level(start_time,end_time,record, level_count)[0]
  File "C:\Users\HP\Desktop\maths\project\djangochar\app\objects.py" in get_max_score_of_level
    59.     return [correct_ans, penalty, time_taken]
  File "C:\Python34\lib\site-packages\django\core\handlers\base.py" in local_vars
    14.     locals = dict(locals)

Request information
GET      No GET data
POST     No POST data
FILES    No FILES data
COOKIES Variable Value
sessionid "ca97ed94a2a85a48f85d4e14c17c1928"

```

Figure: Bug ID 9

9.6.2.3 Gender Options

Bug ID 10: In the Register User Section, the Gender attribute was a text field and it accepted any input.

Gender:	Shubham
School name:	

Figure: Bug ID 10

9.6.2.4 Unrealistic DOB

Bug ID 11: In the Register section, the Date of Birth was not validated and it accepted future dates as well as dates that are not feasible (e.g.: year of birth: 1000)

The screenshot shows a registration form with the following fields:

- Name: Shah
- Date of Birth: 3000-04-06
- Gender: Male

The 'Date of Birth' field contains the value '3000-04-06', which is an invalid date as the year 3000 has not yet occurred.

Figure: Bug ID 11

9.7 Manual Testing Results

BUG ID	CURRENT STATUS	Reason
1	Resolved	-
2	Resolved	-
3	Unresolved	Three.js Library Bug
4	Resolved	-
5	Resolved	-
6	Resolved	-
7	Unresolved	

8	Resolved	-
9	Resolved	-
10	Resolved	-
11	Resolved	-

9.8 Automated Testing Results

Automated Testing was done using selenium web browser. Following were the test results:

Feature	Test Result File	Code Snippet
<u>User Registration</u>	userregres.xlsx	userreg.java
<u>User Login</u>	userlogres.xlsx	userlogin.java
<u>Update User Profile</u>	userprores.xlsx	updateuserpro.java
<u>Selection of Content Category</u>	gameres.xlsx	game.java

10. User Manual – Game Engine

10.1 Description

The goal of this section is to teach a game developer, how to use the game engine and develop amazing games. We will accomplish it by following a step by step tutorial for developing a small game which will touch all the basic steps.

10.2 Before we start

Here is a screenshot of what we are going to develop, the game is about teaching addition using candies.

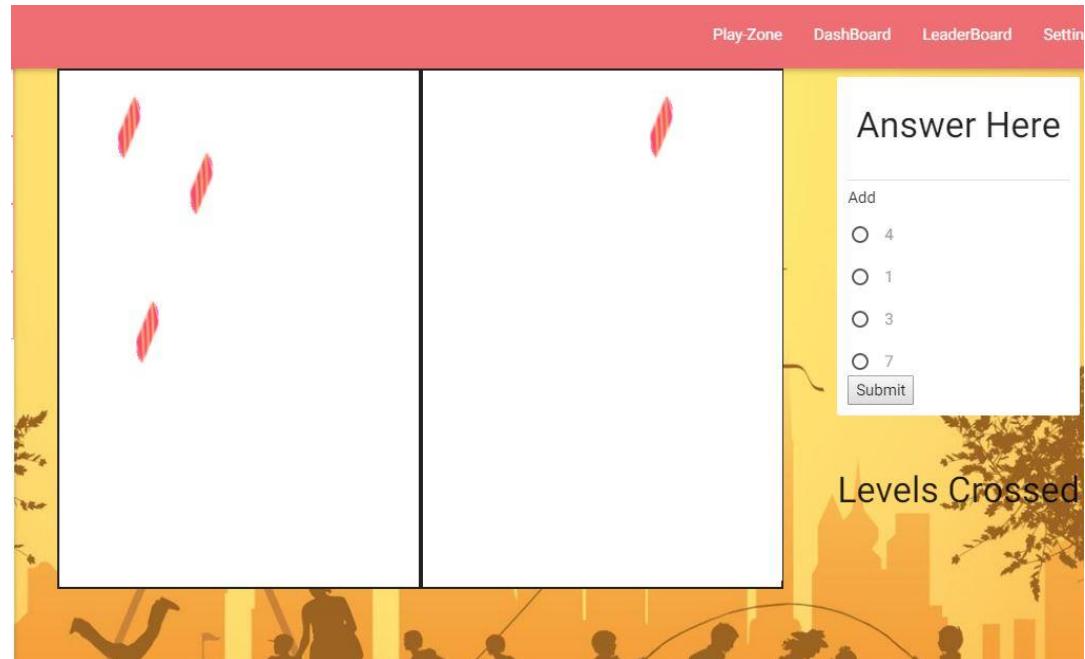


Figure: Candies Addition Game

A game is a blend of theme and subject. Theme is responsible for creating the scene, the objects on the scene, while their behaviour is governed by a subject.

There is one more entity, the user input form, which is essentially responsible for rendering the input form for a game. Needless to say, it is subject specific, ie. , each subject will have a user input form associated with it.

The four entities, theme, objects, subject and forms are stored as JSON Files and contains a complete description of each. In the next section, we teach how to create these JSON Files

10.3 Creating JSON Files

JSON files are the brain of the game engine. They direct the flow of execution of functions, and give necessary description about the number of levels, the assessment counts, user inputs and various other entities that are required to run a game engine.

Let's create the JSON Files

10.3.1 Editing the Empty JSON Files

10.3.1.1 Open the Empty JSON files placed in the dummyJSON folder of the directory.

 .idea	30-Jun-16 11:53 AM	File folder
 __pycache__	30-Jun-16 11:53 AM	File folder
 app	30-Jun-16 11:53 AM	File folder
 djangocharm	30-Jun-16 11:53 AM	File folder
 dummyJSON	30-Jun-16 10:20 PM	File folder
 media	30-Jun-16 6:43 PM	File folder
 static	30-Jun-16 11:53 AM	File folder
 subject	30-Jun-16 6:52 PM	File folder
 templates	30-Jun-16 11:53 AM	File folder
 db.sqlite3	30-Jun-16 9:57 PM	SQLITE3 File 66 KB
 manage.py	08-Jun-16 9:49 PM	Python File 1 KB
 requirements.txt	30-Jun-16 1:58 AM	Text Document 1 KB

Figure: Directory Structure

10.3.1.2 All the three JSON files have been fully described for your reference.

```

1 //JSON File Format for a Subject Creation
2 //Define everything inside the subject variable
3 //Theme = [[functions,assessmentCount],executeCount,rendererCount] as an array of objects
4
5 subject = [
6   [
7     //The Main theme functions are defined here for a level
8     "functions": [
9       [
10         {
11           "returnValue": "", //what the function returns, either a value or a variable name already defined in APILIB
12           "functionName": "", //name of the function that is to be called in the APILIB
13           "params": //The parameters of this function
14             [
15               {
16                 "key": "", //key is the name of the param (not used as such)
17                 "data": //the actual value of the parameter
18               },
19               {
20                 "key": "",
21                 "data": ""
22               }
23             ],
24             //this is a mandatory function, it is to be defined in every subject file
25             "returnValue": "operation",
26             "functionName": "operate",
27             "params": [
28               {
29                 "key": "perform",
30                 "data": //Add the number according to the case number in operate function of APILIB
31               },
32             ],
33             "assessmentCount": 3 //The number of correct answers, the user has to give, for crossing this level
34           ], //add more levels here
35           {
36             "executeCount": 1 //number of times the functions need to be executed to obtain the values
37           },
38           {
39             "rendererCount": 1 //number of renderers required by the game (generally it is equal to the execute count)
40           }
41     ]
42   ]
43 ]

```

Figure: Subject JSON File Format

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

```
1 //JSON File Format for a Form Creation
2 //Define everything inside the form_a variable
3 //form_a = [[questions,types,dynamic,functions,labels]] as objects array
4 form_a =
5 [
6   {
7     "questions": //Define all your questions here as key-value pair
8     [
9       {
10         "key": "",
11       },
12     ],
13     "types": //what type of options are required
14     [
15       {
16         "key": "radio", //radio = radioButton
17         "label": text view
18         //"noSubmit" = no button required
19         //"1" = nothing but plain submit button required
20         "count": 4 //the number of entities required
21       },
22     ],
23     "dynamic": //If functions are required or not
24     [
25       {
26         "key": true //either true or false //if true, define functions below, else define the labels
27       },
28     ],
29     "functions": //Name of the functions to create dynamic options, for each question
30     [
31       {
32         "functionName": "", //Name of the function
33         "params": //the parameters it requires
34         [
35           {
36             "key": "", //key is the name of the param (not used as such)
37             "data": //the actual value of the parameter
38           },
39         ],
40       }
41     ],
42     "labels": //Labels are required if you wish to manually add the options
43     [
44       {
45         "params": [
46           {
47             "key": "" //Define the name here
48           },
49         ],
50       }
51     ]
52   }
53 ]
```

Figure: Form JSON File Format

```
1 //JSON File Format for a Theme Creation
2 //Define everything inside the theme variable
3 //Theme = [functions,obj1,obj2,...] as an array of objects
4 theme =
5 [
6   {
7     "functions": //The Main theme functions are defined here
8     [
9       {
10         "returnValue": "", //what the function returns, either a value or a variable name already defined in APILIB
11         "functionName": "", //name of the function that is to be called in the APILIB
12         "params": //The parameters of this function
13         [
14           {
15             "key": "", //key is the name of the param (not used as such)
16             "data": "" //the actual value of the parameter
17           },
18           {
19             "key": "", //key is the name of the param (not used as such)
20             "data": "" //the actual value of the parameter
21           },
22         ],
23       },
24       {
25         "returnValue": "",
26         "functionName": "",
27         "params": [
28           {
29             "key": "",
30             "data": ""
31           }
32         ],
33       }
34     ],
35     //after functions, a theme may have various objects to be rendered, define them here
36     //The name should be identical to that of the JSON variable of this object
37     //see dummy_object.json to get an idea
38     [
39       {
40         "name": "dummy_object"
41       }
42     ]
43   ]
44 ]
```

Figure: Theme JSON File Format

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

```
1 //JSON File Format for a Theme Creation
2 //Define everything inside the dummy_object variable //You can change the name of this object
3 //Make sure to include it in your theme.json file by this name
4 //dummy_object = [functions] as object
5
6 dummy_object = '
7 [
8     //The Main object functions are defined here
9     "functions" :
10    [
11        {
12            "functionName": "", //name of the function that is to be called in the APILIB
13            "params": //The parameters of this function
14            [
15                {
16                    "key": "", //key is the name of the param (not used as such)
17                    "data": "" //the actual value of the parameter
18                }
19            ] //Define more params here
20        } //Define more functions here
21    ]
22 ]'
```

Figure: Object JSON File Format

10.3.1.3 Open the dummy_theme.json file, we'll edit it first.
Edit the file so that it looks something like the screenshot here:

```
//Delete all the spaces and new line characters before parsing this file
theme = '
[{
    "functions":
    [
        {
            "returnValue":"objectIndex",
            //function needed to randomly choose one of the 4 candies (objects of this theme)
            //the returned value is passed to the randomize() function
            //the theme specific function for 'candies' theme
            "functionName":"randInt",
            "params":
            [
                {
                    "key":"startRange",
                    "data":1
                },
                {
                    "key":"endRange",
                    "data":4
                }
            ]
        },
        //The theme specific function - randomize() - for details, read apilib.js documentation
        {
            "returnValue":"",
            "functionName":"randomize",
            "params":
            [
                {
                    "key":"howManyToDisplay",
                    "data":"correctAns"
                },
                {
                    "key":"objectIndex",
                    "data":"objectIndex"
                }
            ]
        }
    ],
    "name":"candy_a"
```

Figure: Candies Theme JSON File

Add the objects in the theme as follows,

```
[{
  "key": "howManyToDisplay",
  "data": "correctAns"
},
{
  "key": "objectIndex",
  "data": "objectIndex"
}
]
},
{
  "name": "candy_a"
},
{
  "name": "candy_b"
},
{
  "name": "candy_c"
},
{
  "name": "candy_d"
}]
```

Figure: Adding Object Info in Theme JSON File

Save and close the theme file and follow the next steps.

10.3.1.4 Now we'll edit the dummy_object.json file. Edit the file so that it looks something like this screenshot:

```
//Delete all the spaces and new line characters before parsing this file
candy_a = [
{
  "functions" :
  [
    {
      "functionName" : "drawCandy1Random",
      //name of the function that is to be called in the APILIB - rendering the candy
      "params": []
    }
  ]
}]
```

Figure: candy a JSON File

```
//Delete all the spaces and new line characters before parsing this file
candy_b = '
[{
    "functions" :
    [
        {
            "functionName" : "drawCandy2Random",
            //name of the function that is to be called in the APILIB - rendering the candy
            "params": []
        }
    ]
}]'
```

Figure: candy_b.JSON File

Similarly make two more JSON files for objects: candy_c and candy_d. Save and close the files and follow next steps.

10.3.1.5 Our theme is set, let's edit the dummy_subject.json file and dummy_form.json file. Edit these files so that they look something like these screenshots :

```
//Just one level demonstrated
//Delete all the spaces and new line characters before parsing this file
subject = '
[{
    "functions":
    [
        {
            "returnValue":"correctAns", //now add the function in APILIB
            "functionName":"randInt",
            "params":
            [
                {
                    "key":"startRange",
                    "data": 1
                },
                {
                    "key":"endRange",
                    "data": 10
                }
            ],
            {
                //this is a mandatory function, it is to be defined in every subject file
                "returnValue":"operation",
                "functionName":"operate",
                "params":
                [
                    {
                        "key":"perform",
                        "data": 1 //In the switch case under the function 'operate', data will be '1'
                    }
                ],
                "assessmentCount" : 3
            },
            {
                "executeCount":2 //The same process of rendering one number is to be repeated twice, hence count is '2'
            },
            {
                "rendererCount":2 //number of renderers ---> Addition hence '2' numbers in '2' renderers
            }
        ]
}]'
```

Figure: Addition Subject JSON File

```
//Just one level demonstrated
//Delete all the spaces and new line characters before parsing this file
form_a =
[{
    "questions": [
        {
            "key": "Add",
            "label": "Addition"
        }
    ],
    "types": [
        {
            "key": "radio",
            "count": 4 //how many radio buttons, can be changed easily by changing the count
        }
    ],
    "dynamic": [
        {
            "key": "randInt",
            "label": "Random Number"
        }
    ],
    "functions": [
        {
            "functionName": "randInt",
            "params": [
                {
                    "key": "startRange",
                    "data": 1
                },
                {
                    "key": "endRange",
                    "data": 10
                }
            ]
        },
        {
            "label": "Labels"
        }
    ]
}]
```

Figure: Form JSON File

10.3.1.6 The JSON files are ready. Follow the next steps to add the files to target directory make necessary functions.

10.3.2 Add the JSON Files into directory

10.3.2.1 Add the Theme and Object JSON files to dummy_candies folder in Themes folder in the directory.

 demo	30-Jun-16 11:53 AM	File folder
 mineSweeper	30-Jun-16 11:53 AM	File folder
 multipleObjects	30-Jun-16 11:53 AM	File folder
 noTheme	30-Jun-16 11:53 AM	File folder
 numberLine	30-Jun-16 11:53 AM	File folder
 numbers	30-Jun-16 11:53 AM	File folder
 shapes	30-Jun-16 11:53 AM	File folder
 dummy_candies	30-Jun-16 6:57 PM	File folder

Figure 1 Theme Directory Structure

10.3.2.2 Add the Subject and Form JSON files to dummy_addition folder in the Subjects folder in the directory.

 identifyName	30-Jun-16 11:53 AM	File folder
 identifyNum	30-Jun-16 11:53 AM	File folder
 lcmGcd	30-Jun-16 11:53 AM	File folder
 multiplication	30-Jun-16 11:53 AM	File folder
 numLineAdd	30-Jun-16 11:53 AM	File folder
 numLineCo	30-Jun-16 11:53 AM	File folder
 numLinePred	30-Jun-16 11:53 AM	File folder
 numLineSub	30-Jun-16 11:53 AM	File folder
 numLineSuc	30-Jun-16 11:53 AM	File folder
 placeValue	30-Jun-16 11:53 AM	File folder
 primeCompClickable	30-Jun-16 11:53 AM	File folder
 primeComposite	30-Jun-16 11:53 AM	File folder
 subtraction	30-Jun-16 11:53 AM	File folder
 dummy_addition	30-Jun-16 6:56 PM	File folder

Figure: Subject Directory Structure

10.4 Adding Required APIs to the API Library

10.4.1 Open the apilib.js from the js folder which is present in the static folder.

📁 .idea	30-Jun-16 11:53 AM	File folder	
📁 __pycache__	30-Jun-16 11:53 AM	File folder	
📁 app	30-Jun-16 11:53 AM	File folder	
📁 djangocharm	30-Jun-16 11:53 AM	File folder	
📁 dummyJSON	30-Jun-16 10:20 PM	File folder	
📁 media	30-Jun-16 6:43 PM	File folder	
📁 static	30-Jun-16 11:53 AM	File folder	
📁 subject	30-Jun-16 6:52 PM	File folder	
📁 templates	30-Jun-16 11:53 AM	File folder	
📄 db.sqlite3	30-Jun-16 9:57 PM	SQlite3 File	66 KB
🐍 manage.py	08-Jun-16 9:49 PM	Python File	1 KB
📄 requirements.txt	30-Jun-16 1:58 AM	Text Document	1 KB

Figure: Directory Structure

10.4.2 Go through the documentation of APILib

10.4.2.1 The file has been fully described for your reference, it is advisable to go through the file for to get an idea about the coding standards.

10.4.3 Define the Functions

10.4.3.1 Let's define the theme functions and object functions first. The first function is "randomize", you can refer the documentation of randomize which is already present in the API Library.

```

randomize : function(params){
    var n = params[0];
    objectIndex = params[1];
    var obj = JSON.parse(eval(["objectIndex"].name));
    var func = obj[0].functions[0].functionName;
    var width=100;
    var height=100;
    var i = 0;
    objects = [];
    while(i < n)
    {
        var object = window["apilib"][func](obj[0].functions[0].params);
        object.position.x = Math.random() * 1000 - 500;
        object.position.y = Math.random() * 600 - 300;
        var overlapping=false;
        for(var j = 0;j < objects.length;j++)
        {
            var otherObjects[j];
            if((object.position.x>otherObjects[j].position.x||object.position.x<otherObjects[j].position.x)
                ||(object.position.y>otherObjects[j].position.y||object.position.y<otherObjects[j].position.y))
                overlapping=true;
            i++;
            break;
        }
        if(overlapping){
            objects.push(object);
        }
        i++;
    }
    if(rendererCount == s[s.length-1].rendererCount)
        rendererCount = 0;
    for(var i=0;i<objects.length;i++){
        window["scene"+rendererCount].add(objects[i]);
    }
    rendererCount++;
},
objectRenderer : function(params){

```

Figure: randomize function definition

Let's also define the object functions here, which will be called to render the candies.

```

//Functions for rendering the objects
drawCandy1Random : function(){
    var geometry = new THREE.CircleGeometry( 80, 4 );
    var texture = new THREE.TextureLoader().load( "../media/images/candy1.png" );
    var material = new THREE.MeshBasicMaterial( { map : texture } );
    var circle = new THREE.Mesh( geometry, material );
    return circle;
},
drawCandy2Random : function(){
    var geometry = new THREE.CircleGeometry( 80, 4 );
    var texture = new THREE.TextureLoader().load( "../media/images/candy2.png" );
    var material = new THREE.MeshBasicMaterial( { map : texture } );
    var circle = new THREE.Mesh( geometry, material );
    return circle;
},
drawCandy3Random : function(){
    var geometry = new THREE.CircleGeometry( 80, 4 );
    var texture = new THREE.TextureLoader().load( "../media/images/candy3.png" );
    var material = new THREE.MeshBasicMaterial( { map : texture } );
    var circle = new THREE.Mesh( geometry, material );
    return circle;
},
drawCandy4Random : function(){
    var geometry = new THREE.CircleGeometry( 80, 4 );
    var texture = new THREE.TextureLoader().load( "../media/images/candy4.png" );
    var material = new THREE.MeshBasicMaterial( { map : texture } );
    var circle = new THREE.Mesh( geometry, material );
    return circle;
},

```

Figure: Object Functions

10.4.3.2 Let's define the subject functions and update the Operations and Calculate functions, which are necessary for generating the options and assessing the user input.

```
randInt : function(params){  
    var min = params[0];  
    var max = params[1];  
    return Math.floor(Math.random() * ( max - min + 1 )) + min;  
},
```

Figure: Functions for generating Random Options

```
operate : function(operation){  
    switch(operation[0]){  
        case 0:  
            return "counting";  
        case 1:  
            return "add";  
    }  
},
```

Figure: Operate Function

```
calculate : function(correctAnsArray,operation){  
    switch(operation)  
    {  
        case "counting":  
            break;  
        case "add":  
            var num = correctAnsArray[0] + correctAnsArray[1];  
            correctAnsArray = [];  
            correctAnsArray.push(num);  
            break;  
    }  
    return correctAnsArray;  
},
```

Figure: Calculate Function to calculate correct answer

10.5 Update the Compatibility Matrix in Admin Panel

10.5.1 Compatibility Matrix

10.5.1.1 Update the theme name and subject name in the compatibility matrix (choose brain if it is a logical game else

mathematics), in our case we will update the mathematics compatibility matrix.

The screenshot shows the 'Speel Admin Panel' interface. At the top, there's a dark blue header bar with the title 'Speel Admin Panel'. Below it, a white section titled 'Site administration' contains a 'APP' section with four items: 'Compatibility Matrices (Brain)', 'Compatibility Matrices (Math)', 'Subjects', and 'Themes', each with 'Add' and 'Change' buttons. Below this is an 'AUTHENTICATION AND AUTHORIZATION' section with 'Groups' and 'Users', also each with 'Add' and 'Change' buttons.

Figure: Compatibility Matrix

10.6 Run the game

The game is ready to be played, run the platform and start playing.

10.7 Contribute

Being Open Source, the apilib is open to contributions, clone and fork us on github at:

github.com/moovon/IITBProject

11. User Manual - Platform

11.1 Login

Users who have previously registered for our Web Application must login by:

1. Entering their User Name.
2. Entering their Password.
3. Selecting Login to advance to the next screen and begin using the application.

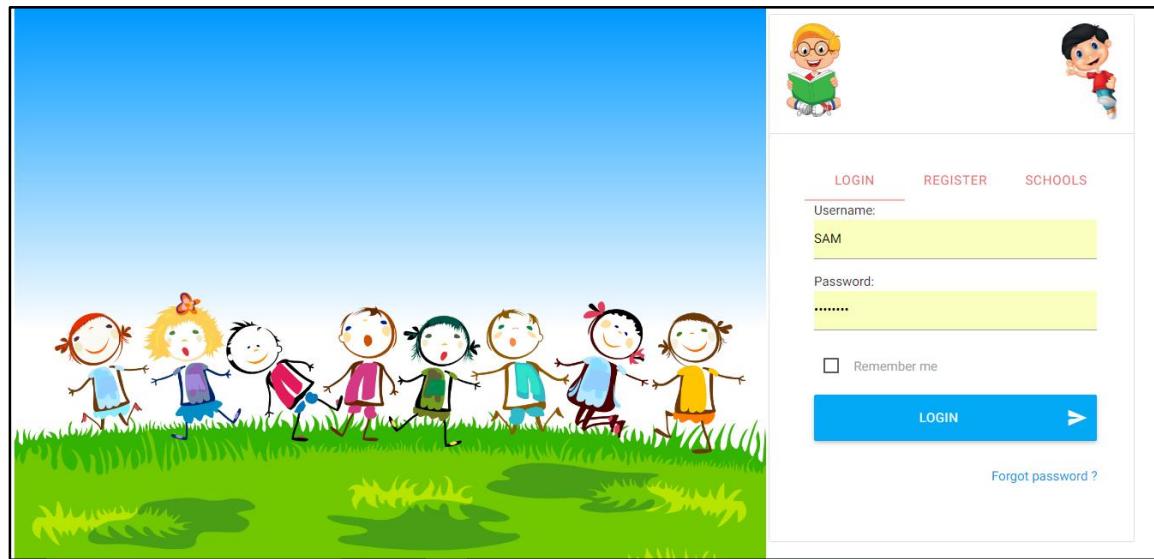


Figure: Login Screen

11.2 User Registration

Users who have not previously registered for the Web Application must select “Register” to access the “New User Registration” from for registration.

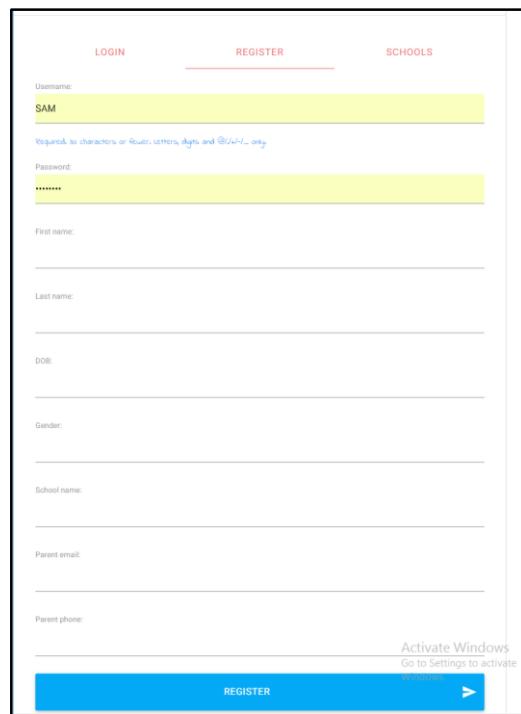
Users will be asked to enter or select the following information –

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

1. Unique username
2. First Name
3. Last Name
4. Password - Passwords must be at least eight characters long and contain at least one letter character and one numeric character. Passwords are case sensitive.
5. Gender
6. School Name
7. Parents Email
8. Parents Phone number

All the information are necessary for user registration

After that they have to click at the activation link send to their email



The screenshot shows a registration form with the following fields:

- Username: SAM (highlighted in yellow)
- Password: (highlighted in yellow)
- First name: (empty)
- Last name: (empty)
- DOB: (empty)
- Gender: (empty)
- School name: (empty)
- Parent email: (empty)
- Parent phone: (empty)

At the bottom right, there is a blue "REGISTER" button with a white arrow pointing right, and a small note: "Activate Windows Go to Settings to activate".

Figure: Registration Screen

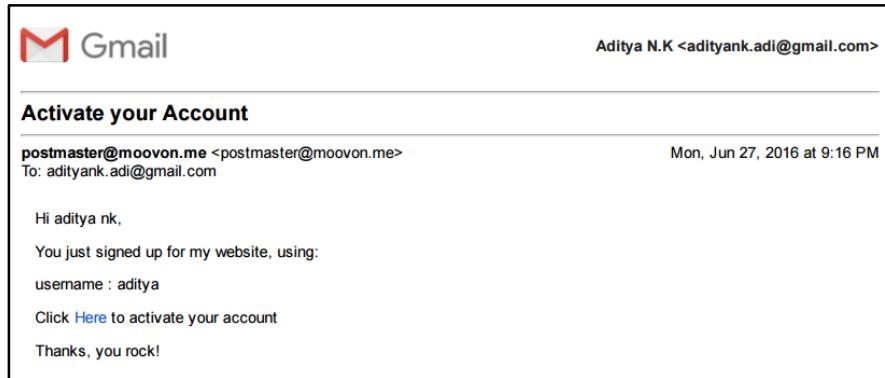


Figure: Email for activation of the account

11.3 Batch Registration

This feature is only available to the admin users.

Users will be asked to enter or select the following information –

1. Admin Username
2. Password
3. Upload Microsoft excel file containing all the details of the kids of a batch.

The screenshot shows an Excel spreadsheet with the following data:

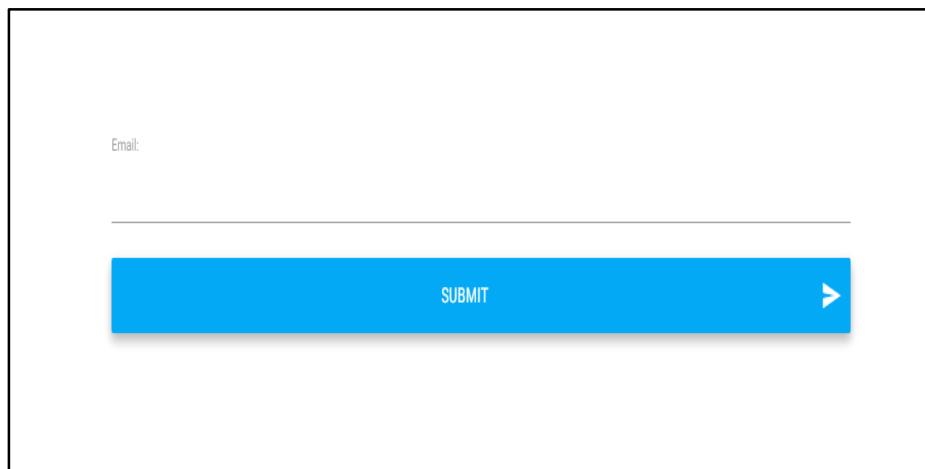
	A	B	C	D	E	F	G	H	I
1	FirstName	LastName	DateofBirth(YYYY-MM-DD)	Gender	School	UserName	password	parent_email	parent_phone
2	Aditya	Tendulkar	1995-2-24	Male	JNV	aditya007	password	example@gmail.co	987654321
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									

Figure: Excel Sheet to upload

11.4 Forgot Password

If a user forgets password they can recover their password by entering their email in the forget password page.

The user then will receive an email to reset their password. And they will have to click the link within two days.



A screenshot of a web form titled "Forgot Password". It has a single input field labeled "Email:" followed by a horizontal line. Below the input field is a blue button with the text "SUBMIT" and a right-pointing arrow icon to its right.

Figure: Email verification for forgot password

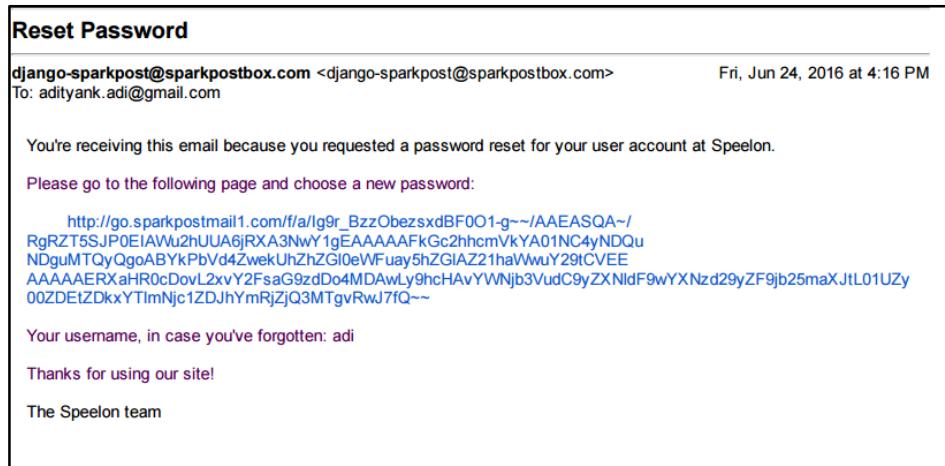


Figure: Email to reset password

11.5 Profile

Profile consists of details of a user and list of games determined by combination of a subject and a theme. There are two separate categories of subject: **Math** and **Brain**.

Math's consists of games according to syllabus of NCERT from class 1 to 5. The games are designed such that it works as the perfect workout to enhance the math skills of a kid. The games are fun and intuitive to play and also, they do not require a deep knowledge of mathematics to play.

Brain's section consists of games that gives the brain of a kid perfect workout. It is a group of memory games, in which the player is faced with outwitting the computer in sound and picture puzzles.

Brain Games utilizes the mouse or touchpad controller. So it is very easy for children to use.

Featuring a total of 25+ games, the catalog of Brain Games included a variety of memory games where the player must focus on a series of symbols, pattern and logic. Each game has several "difficulty switches", which add different aspects of challenge to the user.

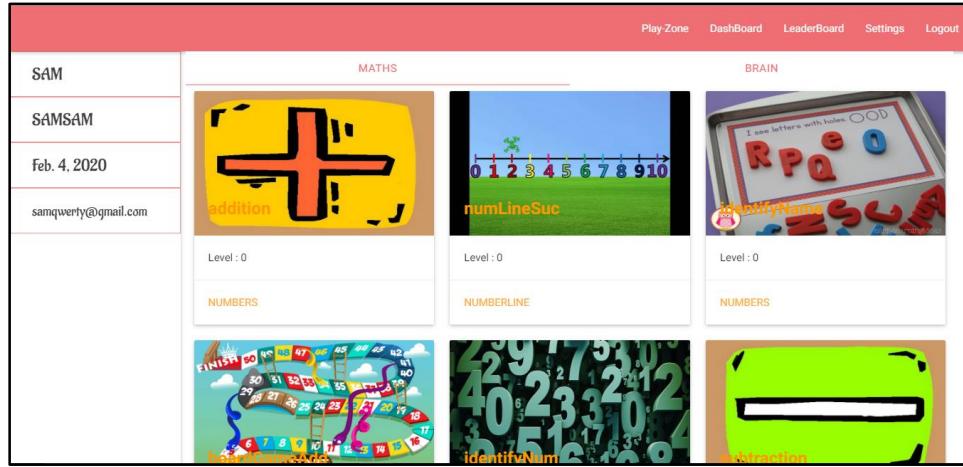


Figure: User Profile

11.6 Update Profile

In this page user will be provided a form containing all the user information which they can change/update according to their wish.

1. First Name
2. Last Name
3. Date of Birth
4. Parents Email
5. School name
6. Parents Phone number

Note: User Name is the only field that cannot be changed.

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

The screenshot shows a user profile update form. On the left, there is a sidebar with the user's name, date of birth, and email. The main area is titled "User Details" and contains fields for Name, Date Of Birth, School Name, Parent's Email, and Parent's Phone. Each field has a text input and a placeholder value. Below the fields are two buttons: "SAVE" and a "Cancel" icon. In the bottom right corner, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Figure: Update Profile

11.7 User Dashboard

User can click at the dashboard tab and there they will find all the games played the user and the levels for each game and scores.

User can click on the subject and they will be redirected to a leaderboard page for that subject only.

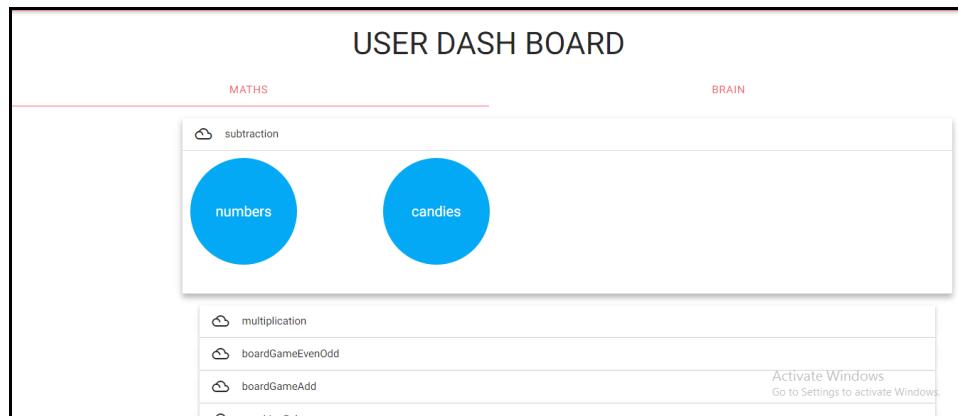


Figure: User Dashboard

11.8 Leader Board

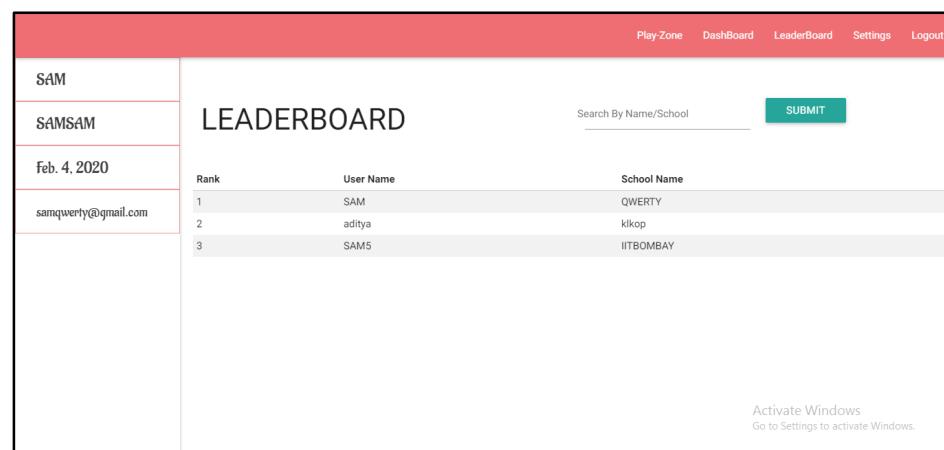
Leaderboard displays the overall rank of all the users.

Leaderboard is a large board for displaying the ranking of the user in tabular form. It displays the place at which the user stands among all the user according to various categories such as in a particular game or at the global scope (among all the registered users).

Leaderboard contains details such as name of user, rank of user, and school name of the user. Other details that can be easily added to leaderboard includes number of correct answer, number of wrong answer, time the user has played games on the platform determined by the user time to answer the question.

Information about contents of leaderboard

1. Rank
2. User Name
3. School Name



The screenshot shows a web-based application interface. On the left, there is a sidebar with the following user information:

- SAM
- SAMSAM
- Feb. 4, 2020
- samqwerty@gmail.com

The main content area is titled "LEADERBOARD". It features a search bar with the placeholder "Search By Name/School" and a "SUBMIT" button. Below the search bar is a table with the following data:

Rank	User Name	School Name
1	SAM	QWERTY
2	aditya	kikop
3	SAM5	IITBOMBAY

At the bottom right of the main content area, there is a small message: "Activate Windows Go to Settings to activate Windows."

Figure: Leaderboard

11.9 Admininstrator rights

Sign in as administrator through the login page will be redirected to admin page

Admin can modify the following things:

1. Compatibility of math subjects with themes
2. Compatibility of math subjects with themes
3. User

Compatibility Matrices (Brain)

Mapping of theme name with subject name

Admin can perform this actions:

1. Updating the existing theme name and subject name record
2. Deleting the existing theme name and subject name record
3. Adding the existing theme name and subject name record

Compatibility Matrices (Math)

Mapping of theme name with subject name

Admin can perform this actions:

1. Updating the existing theme name and subject name record
2. Deleting the existing theme name and subject name record
3. Adding the existing theme name and subject name record.

Admin can perform this actions:

1. Updating the existing theme name and subject name record
2. Deleting the existing theme name and subject name record
3. Adding the existing theme name and subject name record



Figure: Admin Panel

11.10 How to install and run the application

1. Copy/Download the provided exe file.
2. Extract it.
3. The following packages are needed to be installed:
 - a. Python3.5
Linux usually comes with python (make sure its version is 3.5)
<https://www.python.org/downloads/>
(Make sure you tick 'pip' while installing python on windows)
 - b. Django
pip install django or
<https://docs.djangoproject.com/en/1.9/topics/install/>
 - c. Mongodb

<https://docs.mongodb.com/manual/installation/>

d. Mongoengine

pip install mongoengine or
<http://docs.mongoengine.org/guide/installing.html>

e. Requests

pip install requests

f. Sparkpost

pip install sparkpost

g. Pytz

pip install pytz

h. Xlrd

pip install xlrd

Alternatively, all the python packages to be installed through pip can also be installed directly by using the command:

pip install -r /path_to/requirements.txt

TO RUN THE APPLICATION:

1. Open terminal for linux or CMD for windows and type "mongod"(without quotes) and press enter to run the mongodb.
2. Open another terminal (linux) / CMD (windows) and type "python manage.py runserver" (without quotes).

3. Go to a browser (Mozilla/chrome preferred).
4. Open the URL localhost:8000/app
5. The application is ready to use!

12. Challenges

Following were the challenges faced by the team during the development of the platform:

- We cannot safely assume that our user will be able to understand the words, graphics etc. Thus making the platform simple was our main challenge.
- Three.js is a new library and is still in its beta phase, thus certain functions and features that were desired, were too hard to implement.
- Being a single threaded language, it was not possible to stop Javascript for user inputs, in loops, thus we had to find other alternative for that.
- Neither of the team members had experience with web technologies, thus it was a challenging task to develop a platform from scratch.
- Making the game engine configurable was very challenging.
- Making the functions generic enough to work with many themes and subjects was another very important and challenging task.

13. Future Scope

- Being a very basic platform, we plan to further develop it and make it more attractive and interactive.
- There is still scope for development in the game engine, in terms of user input, provision of input by text field is still not present. A lot of functions need to be added into the APILib to make it more generic.
- Group learning feature is to be incorporated into our platform.
- AI Engine to give intelligent suggestions about subjects and difficulty level based on the user performance.
- Develop an executable version of the game engine to remove its dependency on internet.

14. Response and Feedback

Our team went to **Powai English High School, Powai** to demonstrate our platform. We are very happy to have an excellent response from students. The students found our activities very interesting. Similarly, the teachers also gave a very good feedback and found our platform helpful to teach the students in an interactive and efficient way. The scanned Copy of the Appreciation Letter given by the school:

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

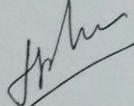


The following students have worked on a project "E3Store-Maths and Brain Assessment Toolset", under **Ekalavya Summer Internship 2016, IIT Bombay**. They have presented a demo of their platform which is aimed at making learning interactive for kids of age group 4-9 Years, here at **POWAI ENGLISH HIGH SCHOOL, POWAI on 1st July 2016**

1. NandanSukthankar
2. Manveer Singh
3. Sunny Mazumder
4. Aditya N.K.
5. Utkarsh Jain
6. Shefali Gupta
7. Saurav Kumar
8. Mayank Arya
9. Shubham Shah
10. Srinidhi Bhat

The demo lesson presented by them in the mathematics and brain games were very useful, attractive and well explained. The children could understand the mathematical and logical concepts in a playful method. The primary teachers appreciated the efforts made by this group.

I congratulate the entire team for their efforts.



HEADMISTRESS

POWAI ENGLISH HIGH SCHOOL, POWAI

Head Mistress

POWAI English Primary School

Mumbai - 400 076.



Powai English High School
Opp. I.I.T. Main Gate, Powai, Mumbai - 400 076.
Maharashtra, India.
Tel. : +91 22 2578 5531
Fax : +91 22 2577 3994
Email : powaienglish@gmail.com
Website : wwwghpeducations.com

PROJECT REPORT – E3STORE-MATH AND BRAIN ASSESSMENT TOOLSET

Figure: Recommendation Letter by PEHS, Powai



Figure: Group Photograph with the kids at PEHS, Powai



Figure: Interaction with the kids



Figure: Kids playing on our platform

15. References

- StackOverflow
- Threejs Official Documentation
- Django Official Documentation
- Python Official Documentation
- MaterializeCSS Official Documentation

16. Glossary

- LTI – Learning Tool Interoperability
- VLE – Virtual Learning Environment
- API – Application Programming Interface
- Compatibility Matrix – A list of themes and subjects depicting which subject is compatible with which theme
- APILib – The central API Library for the game engine
- PIP – Python Installation Package
- Django – A python web framework
- ThreeJS – Lightweight WebGL Library