



**Asim Ghani**

# Digital Cricket Training via Gamification

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

17 May 2021

## Abstract

Author: Asim Ghani  
Title: Digital cricket training via gamification  
Number of Pages: 29 pages  
Date: 17 May 2021  
Degree: Bachelor of Engineering  
Degree Programme: Information Technology  
Professional Major: Mobile Solutions  
Instructors: Patrick Ausderau, Senior Lecturer

---

The purpose of this project is to gamify the learning of Cricket in Finland. Cricket is a sport, which is growing in popularity, and the new learners can struggle with the complex nature of the game. The project aims to provide a digital platform for potential Cricket enthusiasts to learn the game efficiently.

The project was carried out by developing a cross platform mobile application. The application was developed using MongoDB, Express, React Native, Node JS (MERN) stack. The application utilizes the extrinsic motivation technique of gamification.

The project resulted in production of a cross platform mobile application. The application allows the user to begin with by choosing any of the Cricket skill set, namely Batting, Bowling, and Fielding etc. The application also checks the users' knowledge using appropriate tests.

To conclude, it can be stated that this application serves as a platform for eLearning of the sport of Cricket. The application is ready to be tested with learning of Cricket via books and lessons. The comparison can highlight important details about how digital training compares to the traditional teaching of Cricket.

Keywords: Cricket, sports, eLearning, gamification, digital learning

# Contents

## List of Abbreviations

1	Introduction	1
2	Theory	3
2.1	Laws of Cricket	3
2.2	eLearning	4
2.2.1	Types of eLearning	5
2.2.2	Educational applications and games	6
2.2.3	History of educational games	6
2.2.4	Concept of gamifications	7
2.3	MERN Stack	8
2.3.1	MongoDB	8
2.3.2	Express JS	10
2.3.3	React Native	10
2.3.4	React Native and comparison with Flutter	11
2.3.5	Node JS	12
2.4	Tools and third party plugins	13
3	Designing and implementing Cricket learning application (eCricket)	16
3.1	Designing the eCricket application	16
3.2	Implementation of eCricket application	19
3.2.1	Setting up the working environment	20
3.2.2	Custom Components	22
3.2.3	Backend	23

3.2.4	Database Schema	24
3.3	Postman	25
4	Conclusion	26
	References	27

## List of Abbreviations

API:	Application-programming interface
ADB:	Android Debug Bridge
AWS:	Amazon Web Services
CLI:	Command Line Interface
DB:	Database
eLearning:	Electronic learning
HTML:	HyperText Markup Language
HTTP:	HyperText Transfer Protocol
JDK:	Java Development Kit
JS:	JavaScript
JSON:	JavaScript Object Notation
JWT:	JSON Web Token
LBW:	Leg before wicket
MEAN:	MongoDB, Express, Angular, Node JS
MERN:	MongoDB, Express, React, Node JS
MEVN:	MongoDB, Express, Vue, Node JS
MongoDB:	Mongo Database
NoSQL:	Not only Structured Query Language
NPM:	Node Package Manager
OD:	One day
RDBMS:	Relational Database Management System
T20:	Twenty twenty

UI: User Interface

URL: Uniform Resource Locator

XHR: XMLHttpRequest

## 1 Introduction

Cricket is the second most popular sport in the world. It is played by millions and enjoyed by billions of people from all corners of the globe. In 1844, the first international Cricket match was played between the United States of America and Canada. It has continued to grow since then, becoming the lively and exciting modern game that it is played today. Cricket is a bat and ball sport in which two teams of eleven players on each side are involved. The aim of the match is to score more runs than the opposition. One player, the bowler, bowls a hard leather ball that can reach speeds of up to 90 miles per hour, while the batsman, from the opposing side, is ready to hit it with a bat in order to score runs. Cricket is usually played on a circle shaped field and in the middle of that field is a square shaped strip where most of the action takes place is called the pitch. Wickets that are made of three wooden stumps with a support of two bails resting on top of them are found at either end of the pitch. The bowling team has all eleven players on the field at the time of play while the batting team is allowed to have only two batsmen, one on each end of the pitch, at a time. When one batsman gets out, he leaves the field of play and the other batsman walks in to replace him. Once all the batsmen of the batting team get out, it is now the bowling team turns to bat. The team with the most runs after both sides have batted wins. [1; 2; 3]

Electronic Learning (eLearning) is a computer based education system that allows learning anywhere and at any time. The word eLearning refers to the delivery through digital tools of education and training. Technology has changed so far that using such tools has bridged geographical gaps and has created a classroom environment from a distance. [4]

Gamification is using game based mechanics, aesthetics, and game thinking to engage people, motivate action, promote learning, and solve problems [5].

Gamification is a new concept. It involves game elements into non-gaming activities to motivate participants. Each time a student receives a reward, it motivates him/her to do more. It also makes the learning fun, engaging and makes the students spend more time on learning. This is how the strategy of gamification works in education and corporate settings [6].

The software application developed for this thesis will provide digital training for Cricket. The thesis addresses the research question that is the learning of Cricket through digital methods more effective than traditional methods. The application is targeted towards the beginners who want to play or enjoy watching cricket. This study will also explore if the strategy of gamification can simplify the learning experience of the people interested in cricket.



## 2 Theory

### 2.1 Laws of Cricket

The aim of the game is to score more runs than the opposing team. Each team consists of eleven players and both teams are allowed to have an extra player, the twelfth man, in case of injury replacement. At the beginning of the game a coin is tossed between the two captains to decide either their team wants to bowl or bat first. The bowling team also known as the fielding team will put all their eleven players on the field. The batting team only sends out two batsmen at the time of play and they have to bat in pairs since one batsman cannot bat alone. The job of the batsman is to score as many runs as possible within the limited amount of time or the number of overs in case of One Day (OD) and Twenty Twenty (T20) and defend their wicket. One over consists of six legal bowls. In an OD game, each team can bat up to the maximum of 50 overs and in T20, twenty overs only. The wickets are three wooden stumps with two wooden bails resting on top of them. The batsman try to hit the ball away from the wicket with the bat and run to their partner's end of wicket called crease. In order to score one run, a batsman hit the ball and both men ran to the opposite crease. If the bowler bowls an illegal ball, wide or a no ball, this automatically gives one extra run to the batting team. If the batsman hits the ball and the ball bounces before it reaches the boundary rope, this scores four runs and if the ball crosses the boundary rope without bouncing on the ground, this scores six runs. This is the highest scoring shot a batsman can hit on one ball in cricket. [3; 7]

The fielding team job is to restrict the batting team flow of runs and get their batsmen out as quickly as possible. The specific player, bowler, bowls the ball towards the batsman. The bowler has to deliver a legal bowl to be counted towards the game. A batsman can get out in ten different ways in cricket [7].

- Bowled out: If the bowler bowls the ball, hits the batsman's wicket, and knocks off the bails, he has been bowled out.
- Caught out: If the batsman hits the ball in the air and the fielder catches it, he has been caught out.

- Run out: If the batsman runs for his partner's crease and the ball is thrown into the wicket before the batsman gets there, he has been run out.
- Leg Before Wicket (LBW): If the ball hits the batsman leg and the umpire thinks the ball would have hit the wicket if his leg were not in the way, he is ruled out by the LBW [8].
- Stumped out: If the batsman swings and misses the ball, the wicket keeper can catch the ball and push the ball into the wicket. If this happens before the batsman can return to the crease, he has been stumped out.
- Hit wicket: A batsman can get himself out or his own partner out but by accidentally hitting the wicket himself or herself. This is a rare way of getting out in cricket.

There are four more ways of getting out in cricket, which almost never happened in today's game. These include:

- Hitting the ball twice in one stroke.
- Handling the ball.
- Obstructing a fielding player.
- Taking too long to take to the field.

Once the batsman is out, the next batsman in the batting team line up replaces him. Once ten players are out, the players switch sides of batsman and out fielders and vice versa. The highest run total after both sides have had their turn to bat wins. That is a game of cricket in a nutshell but there are a few other things that need to be understood.

There are mainly three different formats of play in Cricket. The amount of overs depends on the format of the game. In T20, each team is given 20 overs for one inning. This game generally lasts about three hours. In OD cricket, each team is given fifty overs for one innings, the game generally lasts about one day. In Test cricket, there are no limits for overs and the game ends when all batsmen are out. It is usually played in two innings. It lasts up to five days and it is the oldest and longest form of cricket [3].

## 2.2 eLearning

The term eLearning refers to training that takes place on a digital device and involves activities such as viewing an instructional video, reading an interesting post, and taking a quiz. It is like contrasting e-book formats to paper books when it comes to eLearning versus conventional education methods. ELearning cannot replace full-scale education, just as digital texts cannot replace the authenticity of a real book. It is, however, available digitally and making learning more convenient [9].

The eLearning allows users to take online courses by using their electronic devices such as computers, laptops, tablets and smartphones. Users can take courses from anywhere and at any time of their convenience with eLearning. The courses can be developed and maintained in a way that keeps user engagement e.g. by the help of audio, video, interactive games and simulations, etc.

### 2.2.1 Types of eLearning

The eLearning comes in many different types, styles, models, and options e.g. lesson-based learning, one-on-one learning, group learning, course based learning, self-paced learning, etc. There are two main categories of electronically enhanced or supported learning.

Synchronous learning is an instructor led online learning. This means that there must be another person available at the same time and at agreed upon location or platform. The learner can use computers or other technology when performing face-to-face training. The learner can also use text based synchronous learning with chat programs and instant messaging. Typically, synchronous learning makes use of schedule, online classes or exam sessions to get everyone together at the same time. [10]

Asynchronous learning is a self-studied style of learning and it can happen at any place and at any time. With asynchronous learning, learners can stay in touch with others by using emails, threaded discussion, message boards, discussion groups, and forums. Course information can be found in text format, audio format, video, and even interactive self-paced software. These forms of learning can be highly effective and can create a very immersive experience for students from learning online especially when combined with simulation and game based learning techniques. [10]

Blended learning is somewhere in the middle, a bit of online self-study along with traditional face-to-face instructions. Students learn theory in their own time asynchronously before a certain date and then they go to real life classes or workshops where discussion and application of the theory are done with peers and trainers. Blended learning can make the transition smoother from paper-based training to eLearning [11].

### 2.2.2 Educational applications and games

The rapid growth of the digital gaming industry has an enormous impact on the new generation. The presence of certain characteristics within the games may be beneficial for educational purposes. This has led to an increase in the interest of the educational community in use of games in the field of education.

Games can be used to provide a better way of education, and they can also transform computers into unique learning resources. Moreover, it can be used as a supplement to conventional teaching approaches to enhance the learning experiences while also teaching other skills such as rule-following, problem-solving, interaction, critical thinking, imagination, teamwork, and good sportsmanship [12].

### 2.2.3 History of educational games

In 1970, logo was the first computer game developed for educational purposes. This game has used a turtle shaped cursor to draw lines with the help of mathematics and programming in order to teach students the basics of coding [13].

In 1971, one of the educational games The Oregon Trail was released. Since then, The Oregon Trail could be found in some form in every school in America until today. The goal of this game is to teach students the American geography and history [14].

In 1977, Apple 2 personal computers were released and it changed the educational gaming business and now educational games can also be played at home. Since the invention of CD\_ROM in 1982, more disk space has become available. Many companies have made educational games to help their players learn in a fun way. [13]

Math blaster was developed in 1987 with a goal to make mathematics an interesting subject to learn for kids. To help find their dog, kids were supposed to blast equations in space, with a hope to make them excited about numbers and mathematics. Math blaster did for math and algebra what Carmen Sandiego did for geography. [13]

In 1969 when the internet was introduced, it greatly expanded the possibility of gaming and educational gaming for many years to come [15]. More recently, Nintendo consoles have had successful educational gaming e.g. MathLand, Scribblenauts, and Nintendo Labo.

In the last 30 years, much has changed regarding educational gaming. From the 8 bits survival game “The Oregon Trail” to the “Mind Craft”, educational gaming continues to grow in terms of technology and innovation.

#### 2.2.4 Concepts of gamifications

Over the last few years, there has been a trend with gamification to add points, badges, levels or leader boards as a way to gamify a product. This has become known as the gamification of blueprint. It relies heavily on rewards and competition to motivate users. These elements are easy to add in existing systems. However, focusing too much on gamifying a system may result in trouble because there are a lot more to good gamification than badges, points and leader boards. There are generally two types of motivation: extrinsic motivation and intrinsic motivation that can engage the user to continue with the activity.

Extrinsic motivation is to encourage users to focus on rewards. A user is extrinsically motivated to perform an activity in order to attain a desired outcome or avoid a negative one. This kind of motivation works very well to motivate a user in some context however research suggests that the value of rewards diminish over time. Therefore a bigger and bigger reward is needed to keep a user motivated [16].

Intrinsic motivation is another type of motivation that helps explain why gamification is engaging. Intrinsically motivated user is not motivated just because of the rewards or avoiding punishment but the user finds enjoyment in the activity itself [17].

## 2.3 MERN Stack

MERN stack consists of four different frameworks and libraries i.e. Mongo Database (MongoDB), Express JavaScript (JS), React JS and Node JS, hence the abbreviation MERN. However, this project will be using React Native instead of React for application development. The role of every component of the MERN stack is described in figure 1.

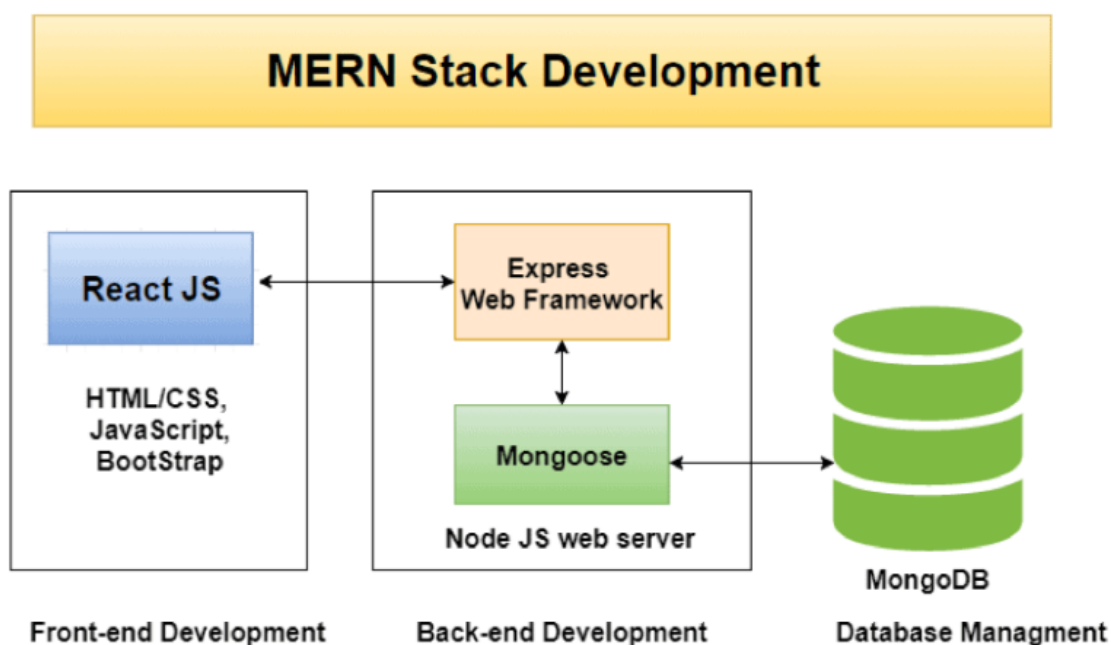


Figure 1. Full stack MERN architecture. Reprinted from [www.medium.com](http://www.medium.com).

MERN is one of the variants of the MongoDB, Express, Angular and Node JS (MEAN) stack, in which React JS replaces the standard Angular JS frontend system. MongoDB, Express, Vue, Node JS (MEVN) is another version, as is any frontend JavaScript application.

### 2.3.1 MongoDB

MongoDB Inc released MongoDB on 11th February 2009. It is an open source, cross platform document oriented Not only Structured Query Language (NoSQL) database. NoSQL does not mean that it has No SQL but it means that it is Not only

SQL database. MongoDB operates on the concept of Documents and Collections [18].

MongoDB uses JavaScript Object Notation (JSON) like files. In MongoDB, data is stored as documents and collections. Each collection can have multiple documents and each document can have multiple different schemas associated with them. In documents, records are stored with data structures containing field and value pairs just like JSON objects but with their own respective schemas. A basic MongoDB document representation is depicted in an example code in listing one.

```
1   {
2       name: "Asim",
3       age: 37,
4       hobbies: ["reading", "painting"]
5   }
```

Listing 1. Example of JSON notation.

As seen in the example code in listing one, on the line 2, `name` is a field and `"Asim"` is a value of type string. In line 3, `age` is a field and `37` is a value of type integer and in line 4 `hobbies` is a field and `"reading"` and `"painting"` are values inside a square bracketed array.

Commonly used terms in MongoDB [19]:

- `_id`: Every document requires an `_id` field. The `_id` field is unique and it can also be interpreted as the primary key in the document. MongoDB will automatically generate an `_id` every time a new document is added.
- **Collection**: A group of many documents in MongoDB is called Collection. It does not require defining rows and columns before creating a collection. In the Relational Database Management System (RDBMS) database, collections can be interpreted as tables.
- **Cursor**: This pointer to the outcome set of a query is a Cursor. The client can iterate through a cursor to retrieve the results.
- **Database**: The location where collections are stored are called Databases, similar to the RDMS database that contains the tables. In physical memory,

every database has its own separate file storage. MongoDB owners can have more than one database at the same time.

- Document: Document is a transcript, which belongs to a Collection. It includes name and value fields.
- Field: Field is a name value pair in a document. The fields are like columns in a RDBMS database.
- JSON: JSON is a lightweight format for transporting and storing data.

### 2.3.2 Express JS

Express JS is a lightweight web framework for Node JS. It is a wrapper for the Node JS HyperText Transfer Protocol (HTTP) module that allows building web servers. Express JS provides extra features to developers for a better programming environment in Node JS. An example code of a basic HTTP server using Express in listing 2.

```
1  const express = require('express');
2  const app = express();
3
4  app.get('/', (req, res) => {
5      res.send('Hello Express');
6  });
7
8  app.listen(process.env.PORT || 3000);
```

Listing 2. An example of a basic HTTP GET request using Express JS.

Express JS makes responding to requests with route support such that developers can write responses to specific Uniform Resource Locators (URLs). It also supports multiple template engines to simplify generating Hypertext Markup language (HTML). It can be installed using Node Package Manager (NPM) and it can be accessed as a required statement in the server files [20].

### 2.3.3 React Native

React Native is a framework for building native applications using JavaScript. It was created by Facebook in 2015. Developers can use their existing JavaScript to build real native apps for iOS and Android.



Developers do not need to learn iOS and Android programming unless they are making a complex application and they need to talk directly with a native Application Programming Interface (API) of the platform. The application code can be written in JavaScript and shared across iOS and Android. That is why many companies these days prefer to build their own applications using React Native because they do not need to hire two separate teams of developers to maintain two different code bases i.e. one for iOS written in Swift or Objective C and another for Android written in Java or Kotlin [21].

The hot reload feature of react native is time efficient [22]. The result can be seen on a physical mobile device or on a simulator as soon as it detects any change in the code base. Whereas in native application development, code needs to be re-compiled every time a developer makes any change and depending on the size of the application, this can take up to several seconds or up to a minute.

A common misconception with React Native is that it can not be used to build a serious application. This can be contradict with applications that are built with this framework such as Facebook, Instagram, Walmart, Bloomberg, etc. [23].

#### 2.3.4 React Native and comparison with Flutter

React Native and Flutter are cross platform frameworks for mobile application development. React Native is created by Facebook and Flutter is created by Google. In terms of programming languages, React Native uses JavaScript whereas Flutter uses Dart, which is a new object oriented programming language created by Google and it is very similar to Java and JavaScript in terms of syntax. Dart is new and growing its community as compared to JavaScript which has been around for a long time and already has a large community of developers. There are lots of books, courses, articles and millions of lines of JavaScript code that can be used in applications.

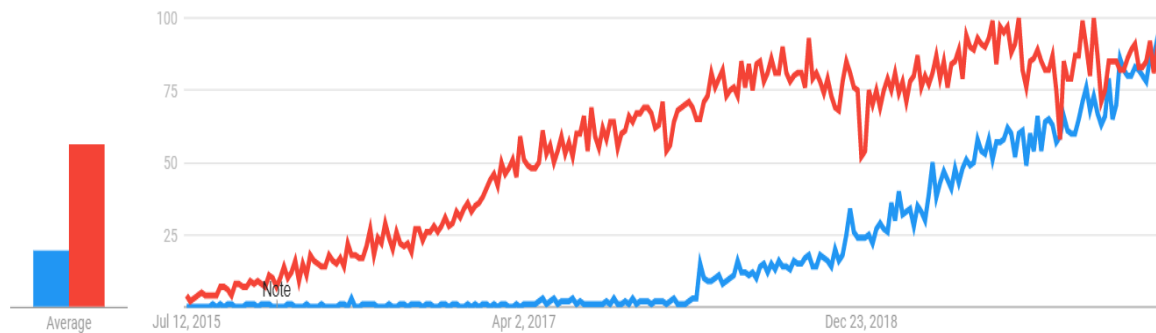


Figure 2: React Native comparison with Flutter in terms of popularity over time.  
Reprinted from [www.radity.com](http://www.radity.com)

In terms of maturity, React Native was launched in 2015 whereas Flutter came out in 2018. Therefore, even though Google has invested a lot in development and improving Flutter, React Native is more mature. Overall, Flutter is a great framework, but React Native has a more adoption rate as of today. The comparison between React Native and Flutter is shown in figure 2.

### 2.3.5 Node JS

Node JS is a runtime environment that allows JavaScript code to be executed outside of a web browser window [24]. The JavaScript is executed using the V8 engine of Google Chrome. It was originally an open source runtime engine of Google, written in C++; similarly, Mozilla and Microsoft use SpiderMonkey and Chakra as their respective runtimes engines for Firefox and Edge browsers.

Node JS also includes tools and functions such as NPM and require, which assist in the management of third party libraries such as Mongoose and Express, allowing for efficient development. NPM is the default package manager that comes with the Node JS environment. These tools also aid in the loading of pre-installed node modules into applications. Node modules are code blocks that can be used anywhere they are required but have no effect on the application code when they are not loaded. Developers may also create their own modules for their own use or for the benefit of the group.

## 2.4 Tools and third party plugins

The application is completed using a number of third party tools and plugins. Even though MERN is a self-contained ecosystem, third party tools and plugins greatly aid in the styling, functionality, and code writing.

Microsoft Visual Studio Code<sup>1</sup> is a common source code editor. It can be used with a number of languages, but MERN creation is simple and productive because it is based on Electron, one of the Node JS frameworks. It is free and open source for developers to use under the MIT license. Web creation is made even better with the built in terminal support [25].

JSON Web Token (JWT) is a JSON web object that aids in the creation of secure data communication between two parties, in this case the server of the app and the end user. A header, a payload, and a signature make up the token. The header specifies how the signature will be calculated [26].

The data contained in the token, such as the user information, is referred to as the payload. To create a signature, the payload is encrypted or hashed. It is used to keep track of who is logging in to the application. The way JWT connects to the application to provide protection and privacy to the contents is shown in figure 3.

---

<sup>1</sup> <https://code.visualstudio.com/>

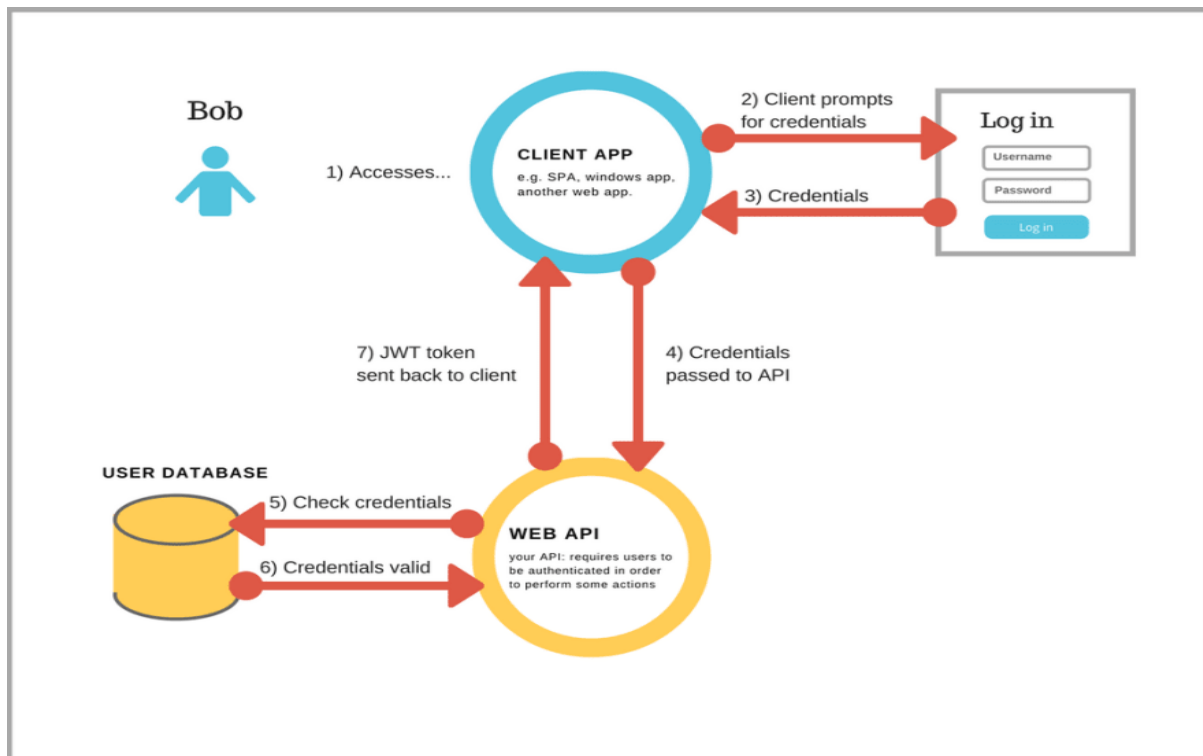


Figure 3. How JWT is generated as part of authentication. Reprinted from [www.dev.to](http://www.dev.to).

Redux is a state management library for JavaScript applications. Since it is just a state management library, it works with React, React Native, VUE, and Vanilla JS. When developing a complex application, developers would almost certainly encounter the need to keep the various parts of the User Interface (UI) in line with one another. If a user modifies data in one section of the UI, all other parts of the UI should be modified immediately to represent the changes. Data will be modified as a result of network requests or background activities in more complicated scenarios. Data can flow from one part of the UI to another in these cases, changing in unexpected ways. To keep everything in sync, developers must write a lot of code, and when something goes wrong, figuring out how the data changed and where it came from becomes difficult. If they run into this issue in an application, it is a sign that they need a state controlled library. [27]

Axios is a JavaScript library that enables HTTP requests to be made on both computers and browsers using Node JS. The majority of the browsers supports it. Axios is a promise-based library that allows asynchronous XMLHttpRequest (XHR)

requests to be created. Axios has a number of advantages over the native Fetch API, such as support for older browsers without the need for polyfilling. It is allowed to cancel a request and set an answer timeout. It supports JSON data transformation and, most notably, Node JS.

### **3 Designing and implementing Cricket learning application (eCricket)**

eCricket is a React Native mobile application that was built during this project with the goal of helping potential cricket players to start learning the sport. The idea for the app came after noticing that many Finnish players are still on the beginner's level of the sport. This application not only teaches how to play the sport but it also helps one to learn the rules of the game in case one is interested in watching the sport.

#### **3.1 Designing the eCricket application**

The application consists of three main screens i.e. Home Screen, Quiz Screen and Profile Screen. Home Screen is the landing page of the application. It displays the different skill sets a user can select and start learning about the Cricket. It is designed by using a card component. Cards display content and actions on a single topic. They should be simple to search for useful details. Text and photographs, for example, should be placed on them in a way that clearly denotes hierarchy. The design of Home Screen is shown in figure 4.

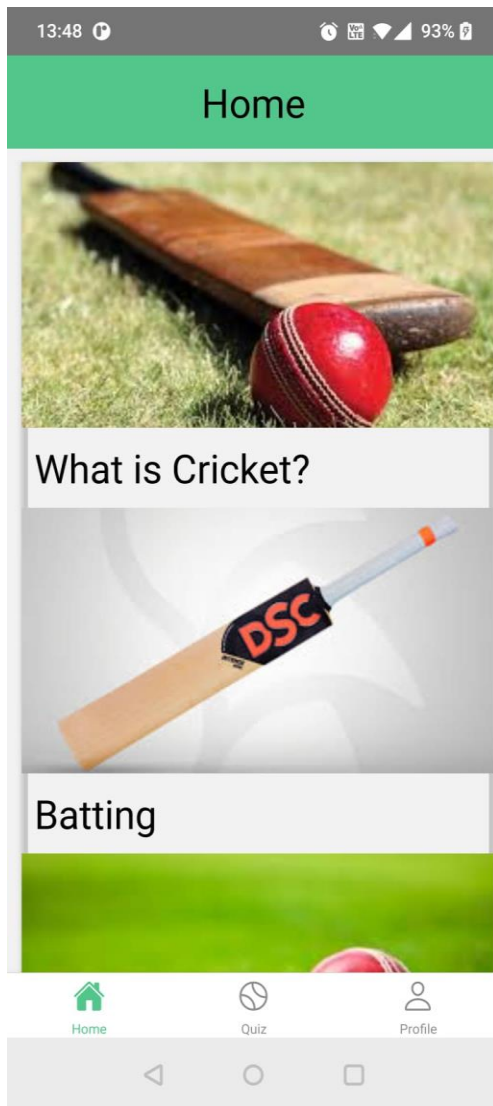


Figure 4: Screenshot of a Home Screen.

Users can tap on the card of their own choice and start answering the questions related to that specific skill set. This sort of technique is called trial and error. The theory behind trial and error is that learners can learn better, when they do the task themselves. Instead of overwhelming the user by lots of reading material, we have decided to start the quiz as a mode of gamification so that the users can learn from their own mistakes and stay extrinsically motivated to use the application.

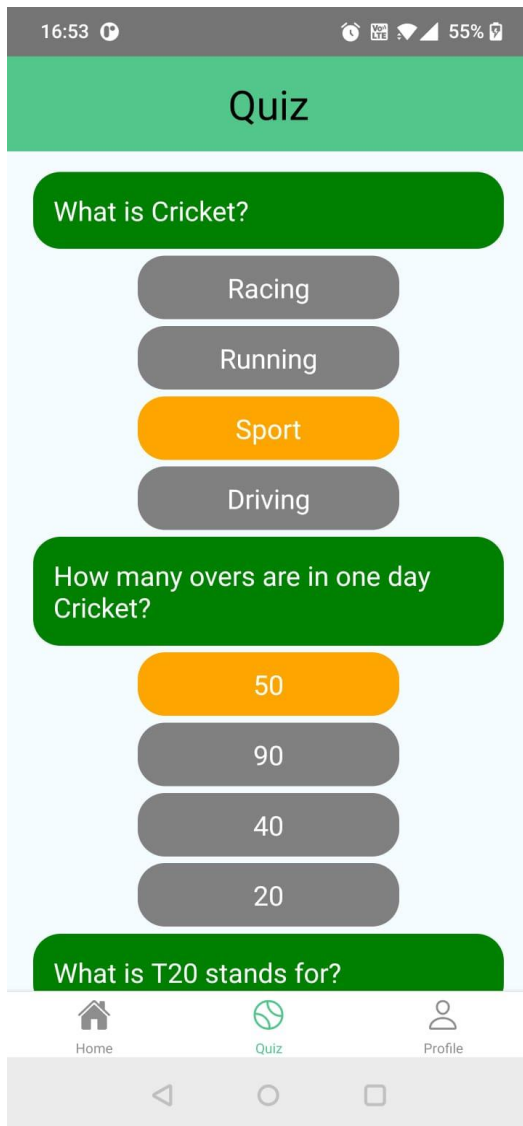


Figure 5. Screenshot of the Quiz Screen.

A user will come to the Quiz screen of the application after selecting a skill set in the Home screen. Quiz screen will start asking questions according to the selected skill. A question with four different possible answers will be asked. Every right answer will give the user some points according to the level a user has already achieved. By collecting the points, the user can earn badges and raise his/her level of the game. The more the user plays, the more points s/he will get and eventually improve their skill level of the game of Cricket. The design of a Quiz Screen is illustrated in figure 5.



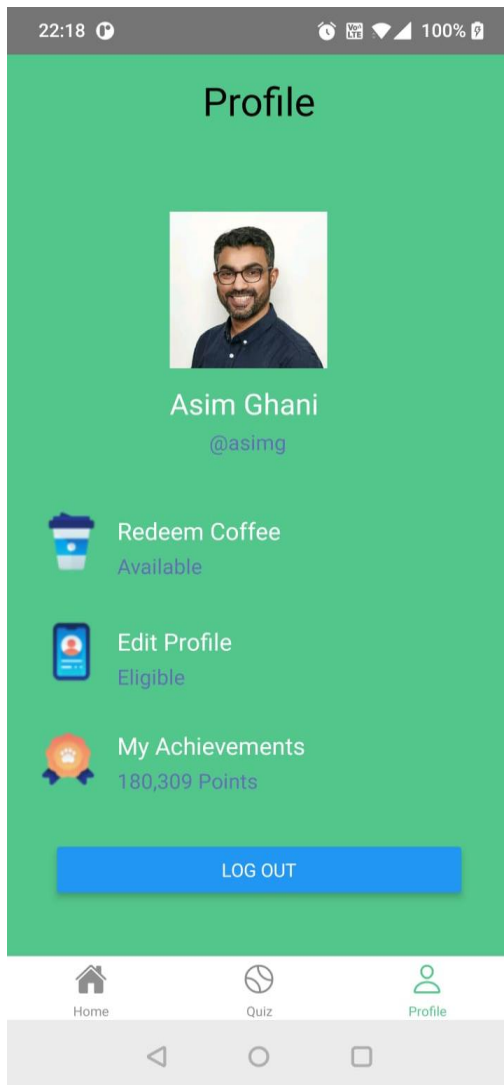


Figure 6. Screenshot of a Profile Screen.

A user can jump on the Profile Screen by tapping on the bottom navigation bar at any time. Profile Screen provides the user detailed information as well as the different voucher and achievements accomplished by the user. Users can collect points by giving correct answers in the questionnaire and hence improved their rank and can get different badges and vouchers. The design of the Profile Screen is illustrated in figure 6.

### 3.2 Implementation of eCricket application

Creating an application requires some planning and execution. In the hope to help people learn cricket digitally, a React Native application has been created. When building the eCricket app these steps were followed.

### 3.2.1 Setting up the working environment

To build an application, it is required to set up the working environment on the local machine. React Native documentation has clear instructions on how to set it up. According to documentation, there are two different ways of running a React Native application, Expo client and React Native Command Line Interface (CLI).

In this application, the process of running React Native applications via its own CLI will be followed. Even though Expo client gives us a quick start but React Native CLI gives us full control on the building process. There are a few basic steps that need to be followed in order to run an application via React Native CLI. The first step is to install Node JS. It can be installed from the official website but React Native official documentation recommends installing it via Chocolatey and Homebrew, a popular package manager for Windows and Mac OS respectively. Either way, Node JS version 12 or higher needs to be installed. React Native also requires a Java Development Kit (JDK), which can also be installed via Chocolatey and Homebrew, and requires JDK version 8 or newer to develop a React Native application.

React Native applications can be run on both Android and iOS at the same time. However, since they do not own iOS devices, the developers focus on the Android version for testing and development. To run a React Native application on a physical device, Android Debug Bridge (ADB) needs to be installed. ADB makes it possible to run/test the application with an emulator or a real device. To use a real device, it has to have the USB debugging enabled.

Windows users need to configure Android home variable on their machines so that every time they run a command from the Windows terminal, the machine automatically connects it with a device connected with Android studio. The environment variable can be set up by following steps [28]. It is also illustrated how to set up the environment variable in figure 7.

1. Open the Windows Control Panel.
2. Click on User Accounts, then click User Accounts again
3. Click on Change my environment variables

4. Click on New... to create a new `ANDROID_HOME` user variable that points to the path to your Android SDK:

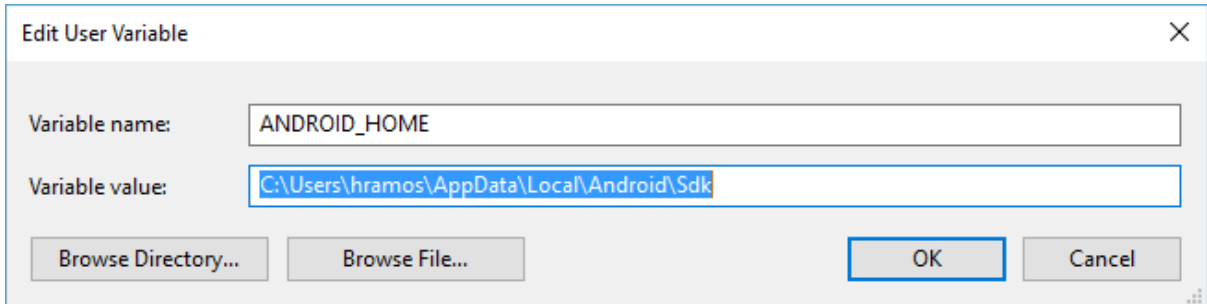
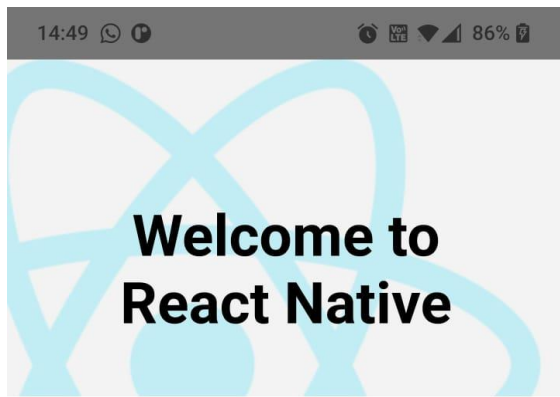


Figure 7. Setting up the environment variable. Reprinted from [www.reactnative.dev](http://www.reactnative.dev)

Once Node JS and the JDK is installed and the phone and `ANDROID_HOME` are configured, a new React Native application can be created by running the `npx react-native init AwesomeProject` command on the terminal. This process might take some time and it will install all the required packages automatically. After the installation process, changing directory into the project folder and running the `npm run android` command will start the newly created React Native application on an Android device.



### Step One

Edit **App.js** to change this screen and then come back to see your edits.

### See Your Changes

Double tap **R** on your keyboard to reload your app's code.

### Debug

Press **Cmd or Ctrl + M** or **Shake** your device to open the React Native debug menu.

### Learn More

Read the docs to discover what to do next:



Figure 8: First view of React Native application.

The application will run on a physical device if everything is completed successfully. The first look of the newly created React Native application on a physical device is illustrated in figure 8.

### 3.2.2 Custom Components

React Native library is written on top of React JS which is created by Facebook. The core concept of React JS is that it separates the whole application into small separate components which work together to build one large application which a client can see and interact with.

```

1   const Header = (props) => {
2   return (
3     <View style={styles.viewContainer}>
4       <Text style={styles.textContainer}>
5         {props.title}
6       </Text>
7     </View>
8   );
9   };
10
11  const styles = StyleSheet.create({
12  viewContainer: {
13    padding: 15,
14    height: 70,
15    backgroundColor: '#53c68c',
16    alignItems: 'center',
17    shadowColor: 'yellow',
18    shadowOffset: {width: 0, height: 2},
19    shadowOpacity: 0.5
20  },
21  textContainer: {
22    fontSize: 30
23  }
24  })
25
26  export default Header;

```

Listing 3. An example of a reusable Header component.

Since React Native is a part of React JS, it follows the same concept and techniques. React Native provides a number of components that can be used to develop the application. Custom components can also be constructed by combining these existing components to meet the application needs. We developed a few custom components for the eCricket application based on the application requirements. An example code of a reusable Header component is in listing 3.

### 3.2.3 Backend

The application needs a backend with which the frontend, or client side, will communicate and support the user according to their needs. Business logic is typically written in backends, and frontends can communicate with APIs from them. Its primary function is to take requests and respond to them.

```

1   app.get("/quizzes", (req, res) => {
2     Quiz.find((err, data) => {
3       if (err) {
4         res.status(500).send(err);
5       } else {

```

```

6             res.status(20).send(data);
7         }
8     });
9 });
10
11 app.post("/quiz", (req, res) => {
12     const quiz = req.body;
13     Quiz.create(quiz, (err, data) => {
14         if (err) {
15             res.status(500).send(err);
16         } else {
17             res.status(201).send(data);
18         }
19     });
20 });

```

Listing 4. An overview of GET and POST API endpoints using Node JS and Express.

In the development of eCricket application, we have used Node JS with Express, which will serve as a backend to this application<sup>2</sup>. An overview of GET and POST API endpoints is in listing 4.

### 3.2.4 Database Schema

The database schema is a blueprint or description of the database model. Before using the database, regardless of whether it is SQL or NoSQL, the schema is declared. However, in the case of NoSQL databases, such as MongoDB, it is not necessary. Data can also be inserted and retrieved without declaring it.

```

1     const mongoose = require('mongoose');
2
3     const QuizSchema = mongoose.Schema({
4         questionNumber: String,
5         question: String,
6         questionOptions: [
7             {option: String}
8         ]
9     });
10
11     module.exports = mongoose.model("Quiz", QuizSchema);

```

Listing 5. An example of a Quiz database schema created with Mongoose.

The eCricket application receives data by calling an API endpoint provided by the backend, which, upon successful completion, requests data from a database and

---

<sup>2</sup> Full source code of the application can be found in <https://github.com/ghaniasim/eCricket>

returns it to the application. The data will be returned in a JSON-like format that was declared as a database schema before it was created. An example of a database schema, which is used in our application, is in listing 5.

### 3.3 Postman

Postman, as the name implies, is an application that can send a message from a client to a server over a network. It is used for developing, testing, sharing, and documenting APIs. Developers can build their own APIs and use it to evaluate them both manually and automatically. They can also share and collaborate on their APIs, and last but not least, they can log them.

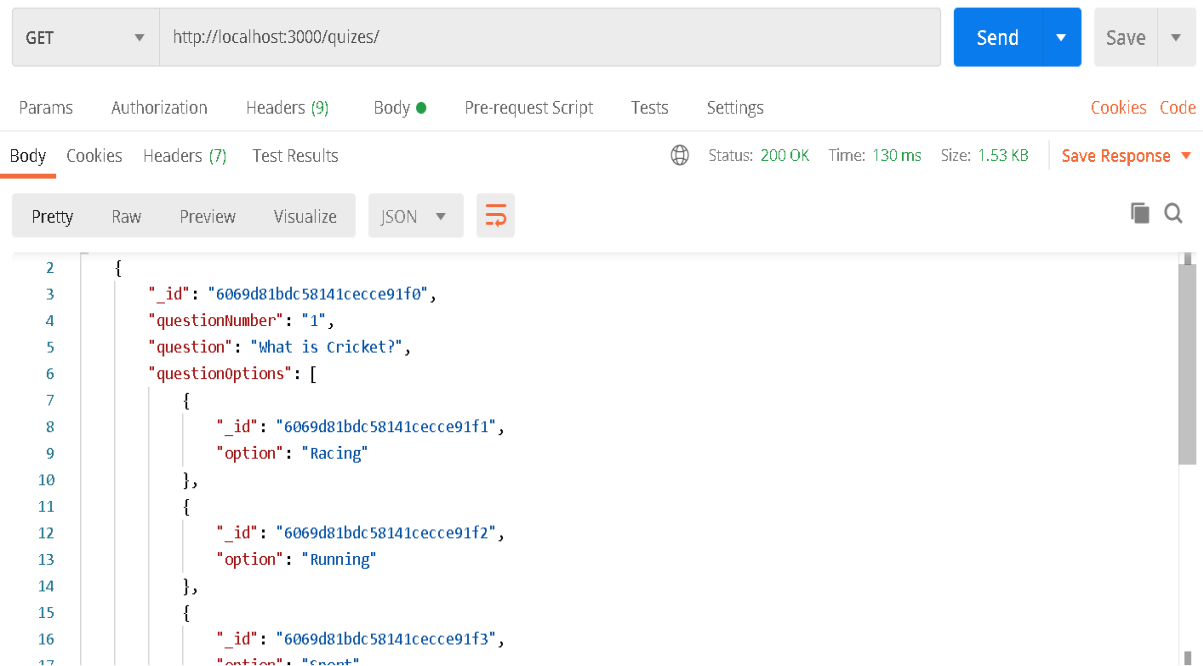


Figure 9. An overview of testing /quizes API endpoint with Postman.

We built our own API endpoints in this eCricket framework, and we wanted to make sure they were returning correct data before putting them into the actual application. We might have used a web browser to test those endpoints, but Postman is the best tool for the job and gives us full control over API testing. An example of using Postman to test the /quizes endpoint with HTTP GET method of the eCricket backend is shown in figure 9.

## 4 Conclusion

To conclude, it can be stated that during the project, a cross platform mobile application that assists the potential Cricket enthusiasts to learn the game in a gamified manner has been produced. The extrinsic motivation technique of gamification has been used to keep the users motivated to use the application. The extrinsic motivation method focuses on the rewards, badges, points and leader board for the users. The application was developed using a number of modern technologies, including MongoDB, Express, React Native, Node JS, React Native CLI and Visual Studio Code integrated development environment.

The application developed in this project has the goal to help to teach the sport to the players from beginner to advanced levels and improve their sporting skills in an interesting way at any time and at any place of their own convenience. Thus, it would be interesting to run a large user test with potential cricket players where half of them will learn from the law of cricket book and the other half using the app in order to evaluate the merits of gamification.



## References

1. International Cricket Council [Internet]. Icc-cricket.com. 2021 [cited 14 April 2021]. Available from: <https://www.icc-cricket.com/about/cricket/history-of-cricket/19th-century>
2. World's Most Popular Sports by Fans [Internet]. Topendsports.com. 2021 [cited 14 April 2021]. Available from: <https://www.topendsports.com/world/lists/popular-sport/fans.htm>
3. The Laws of Cricket | MCC [Internet]. Lords.org. 2021 [cited 14 April 2021]. Available from: <https://www.lords.org/mcc/the-laws-of-cricket>
4. Hubackova S. History And Perspectives Of Elearning. Procedia - Social and Behavioral Sciences. 2015 Apr;191:1187–90.
5. Kapp, Karl M. 1967. The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education. San Francisco, CA: Pfeiffer.
6. Sehoon K. How a company's gamification strategy influences corporate learning: A study based on gamified MSLP (Mobile social learning platform). Elsevier. 2020 Sep18;
7. Cricket Rules and Regulations: ICC Rules of Cricket [Internet]. Cricket Rules and Regulations | ICC Rules of Cricket. [Cited 2021Jan13]. Available from: <https://www.icc-cricket.com/about/cricket/rules-and-regulations/playing-conditions>
8. Adie JM, Renshaw I, Polman R, Thompson MB, Mann DL. When in doubt, it's not out: Match format is associated with differences in elite-level cricket umpires' leg-before-wicket decisions. Psychology of Sport and Exercise. 2020;51:101760.
9. What is E-Learning | IGI Global [Internet]. Igi-global.com. 2021 [cited 10 May 2021]. Available from: <https://www.igi-global.com/dictionary/administrators-assessments-online-courses-student/8785>
10. Lawless C. Synchronous vs Asynchronous Learning: Which is Right for Your Learners? | LearnUpon. 2020 [cited 24 April 2021]. Available from: <https://www.learnupon.com/blog/synchronous-learning-asynchronous-learning/>

11. Rudd KE, Puttkammer N, Antilla J, Richards J, Heffron M, Tolentino H, et al. Building workforce capacity for effective use of health information systems: Evaluation of a blended eLearning course in Namibia and Tanzania. *International Journal of Medical Informatics*. 2019;131:103945.
12. Tierney K. How Technology in the Classroom Is Shaping the Future of Education [Internet]. JotForm. 2021 [cited 10 May 2021]. Available from: <https://www.jotform.com/technology-in-the-classroom/>
13. DE CASTILLA M. History of Educational Games – Playful Learning [Internet]. Commons.pratt.edu. 2018 [cited 18 April 2021]. Available from: <https://commons.pratt.edu/playful-learning/history-of-educational-games/>
14. Porges S. How 'The Oregon Trail' Was Built Without Access To A Computer [Internet]. Forbes. 2018 [cited 18 April 2021]. Available from: <https://www.forbes.com/sites/sethporges/2017/11/27/the-surprising-story-behind-how-the-oregon-trail-was-built-without-access-to-a-computer/?sh=610ed1294f11>
15. An introduction to the Internet [Internet]. TheWebGuild.org. 2021 [cited 24 April 2021]. Available from: <https://www.thewebguild.org/an-introduction-to-the-internet>
16. Extrinsic Motivation: What Is It and How Does It Work? [Internet]. Healthline. 2021 [cited 24 April 2021]. Available from: <https://www.healthline.com/health/extrinsic-motivation>
17. Intrinsic Motivation Theory: Overview, Factors, and Examples [Internet]. Healthline. 2021 [cited 24 April 2021]. Available from: <https://www.healthline.com/health/intrinsic-motivation>
18. What is MongoDB? Introduction, Architecture, Features & Example [Internet]. Guru99.com. 2021 [cited 24 April 2021]. Available from: <https://www.guru99.com/what-is-mongodb.html>
19. MongoDB Terminology - W3Adda [Internet]. W3 Adda. 2021 [cited 24 April 2021]. Available from: <https://www.w3adda.com/mongodb-tutorial/mongodb-terminology>
20. Express routing [Internet]. Expressjs.com. 2021 [cited 4 May 2021]. Available from: <https://expressjs.com/en/guide/routing.html>
21. Why Use React Native For Your Business In 2021? | Resourcifi [Internet]. Resourcifi. 2021 [cited 4 May 2021]. Available from:

<https://www.resourcifi.com/blog/why-use-react-native-for-business/#:~:text=React%20Native%20allows%20cross%2Dplatform,its%20User%20>

22. Introducing Hot Reloading · React Native [Internet]. Reactnative.dev. 2021 [cited 4 May 2021]. Available from: <https://reactnative.dev/blog/2016/03/24/introducing-hot-reloading>
23. Dryka M. 10 Famous Apps Built With React Native [Internet]. Brainhub.eu. 2021 [cited 24 April 2021]. Available from: <https://brainhub.eu/library/react-native-apps/>
24. Kopecky C. What is Node.js? A beginner's introduction to JavaScript runtime [Internet]. Educative: Interactive Courses for Software Developers. 2021 [cited 10 May 2021]. Available from: <https://www.educative.io/blog/what-is-nodejs>
25. Intro to VS Code [Internet]. Medium. 2021 [cited 4 May 2021]. Available from: <https://medium.com/@rithmschool/intro-to-vs-code-7a4caadb7c70>
26. Pham P. Implementing Json Web Token (JWT) to secure your app [Internet]. Medium. 2021 [cited 4 May 2021]. Available from: <https://blog.nextzy.me/implementing-json-web-token-jwt-to-secure-your-app-c8e1bd6f6a29>
27. Redux · An Introduction — Smashing Magazine [Internet]. Smashing Magazine. 2021 [cited 10 May 2021]. Available from: <https://www.smashingmagazine.com/2016/06/an-introduction-to-redux/>
28. Setup Android Environment Variables - AutomationTestingHub [Internet]. AutomationTestingHub. 2021 [cited 10 May 2021]. Available from: <http://www.automationtestinghub.com/setup-android-environment-variables/>