

ABSTRACT

The traditional method of assessment (examination) is often characterized by examination questions leakages, human errors during marking of scripts and recording of scores. The technological advancement in the field of computer science has necessitated the need for computer usage in majorly all areas of human life and endeavors, education sector not excluded. This work, Web-based Examination System (WES) was, therefore, born out of the will to stymie the problems plaguing the conventional (paper-based) examination system by providing a campus-wide service for e-assessment devoid of irregularities and generally fair to examinees and equally enhances instant feedback. This system developed using combination of CSS, HTMLJS., PHP SQL MySQL and Laravel is capable of reducing proportion of workload on examination, grading and reviewing on the part of examiners. Thus, the system enables the release of examination results in record time and without error. WES can serve as an effective solution for mass education evaluation and offers many novel features that cannot be implemented in paper-based systems, such as real time data collection, management and analysis, distributed and interactive assessment towards promoting distance education.

INTRODUCTION

The explosive growth of the Internet is making available radical new means of communication that affect life in diverse areas as business, entertainment and education. While older methods of accomplishing tasks continue to be used, the Internet offers unique advantages (Rahneva, 2004). One important area of application of the web technology is in the development of web-based testing and assessment (Iyilade & Adekunle, 2005). Before the massive influx of Information Technology (IT), students's academic performances were evaluated via paper-based system of assignments and tests. Since recent progress in state-of-the-art IT has advanced significantly, educational products are now available electronically (including the web technologies) and new methods of educational assessment have emerged. 31 The World Wide Web (WWW) has gained popularity within educational sector and become an inexpensive, easily accessible way to communicate, disseminate information, teach and examine courses and conduct researches. Consequently, there exists wide preference and adoption of web-based testing and assessment over the traditional paperbased method of assessment which has over decades been characterized by examination questions leakages, human errors during the marking and recording of scores. Web-based testing and assessment systems offer greater flexibility than the traditional approach because test could be offered at different times by students and in different locations (Akanbi & Adetunji, 2012). e-Examination system rides on the huge success of Information and Communication Technology (ICT) and its various features, security, reliability and consistency. The system simplifies the examination process by computeraided control and automatic marking to reduce the complex paper work (Meng & Lu, 2011). In this current adoption of ICT towards promoting distance education where distance between learners and instructors is no longer a barrier, Web-based Examination System (WES) is an effective solution for mass education evaluation. The system is based on a Browser/Server

framework which carries out the examination and auto-grading for objective questions. The system facilitates conducting examinations, collection of answers, auto marking the submissions and production of reports for the test. It will be used via Internet and is therefore suitable for both local and remote examination. The system developed reduces the large proportion of workload on examination, grading and reviewing. It also has the potentials to reduce drastically examination malpractice as examinees are duly authenticated online in a real-time manner and their results are released some few minutes after the completion of the examination because where the lecturer would spend weeks marking scripts and grading manually, the computer would grade the students as soon as they finish their paper, get their already stored continuous assessment and produce their eventual result. It also enhances effective distance education as examinees can write examination in any part of the world and equally get their results instantly. The remaining parts of the paper is organized as follows. Section 2 reviewed the related literatures. In Section 3, methodology and system design techniques are highlighted. The Section 4 focuses on the implementation of the design while Section 5 presents the conclusion and recommendations.

Traditional versus Electronic Tests In educational institutions, tests are conducted to evaluate the academic progress of learners; review, compare and measure the effectiveness of methods of instruction; serve as basis of guidance and counselling to students, selection for prize award and employment and grading for the purpose of certification. In the context of education system, one possible purpose of a test is to assess whether a learner has attained an educational goal. The outcome of such a test can help instructors analyze problems with his method(s) of instruction and to better understand the learner's strengths and weaknesses in a given subject. Test can be used to fine-tune the instruction environment or method based on the analytical results to improve the instructor's teaching performance (Chen, et. al., 2005). There are currently two methods for conducting tests: (i) The traditional method of using paper and pencil tests, including the creation of test items, the grading of students' test sheets, and the analysis of learner's responses for each test item, which is considered to be tedious; and (ii) Computer-Based Tests, an electronic examination, allows test activities to be carried out using different electronic platform/environment. Basically, the electronic examination (eExamination) system involves the conduction of examinations using various electronic 32 devices (mobile phones, computers etc) connected to the testing system via the Internet or the Intranet. The process is predominantly automated, which means the administration, grading, reviewing of the examination is of little effort. Usually the examination is in form of multiple-choice test. Ayo, et. al. (2007) defines e-examination as a system that involves the conduct of examinations through the web or the intranet. They proposed a model for e-Examination in Nigeria where all applicants are subjected to online entrance examination as a way of curbing the irregularities as proposed by the Joint Admissions Matriculation Board (JAMB), a body saddled with the responsibility of conducting entrance examinations into all the Nigerian universities. This model was designed and tested in Covenant University, one of the private universities in Nigeria. Their findings revealed that the system has the potentials to eliminate some of the problems that are associated with the traditional methods of examination such as impersonation and other forms of examination malpractices. Gardner, et. al. (2002) in their work developed a computer-supported learning system, named CECIL, which included an interesting function of 'self-Assessment' to enhance students' learning effectiveness. The function of 'self-assessment' is equipped with item pools and teachers can administer and construct examinations easily through the Internet. They also pointed out that the advantages of item pools are that 'teachers are able

to incorporate large item banks (item pools) from textbook publishers and batch load these questions with a minimum of manual effort'. Moreover, Gardner et al. (2002) also stated that teachers who administer and construct an examination through the Internet have the advantage of helping students to check their understanding of the learning materials at all hours. Wang et al. (2004) in their work developed an assessment system using Triple-A Model (assembling, administering, and appraising) as the baseline qualification in order to provide the most comprehensive form of Computer-Based Test (CBT) or Web-Based Test (WBT) and to be more suitable for teacher education. The Triple-A Model includes the essential functions of CBT system. Assembling deals with the construction of item pools, test items, and schedules of tests. Administering is to assign the test items and item choices randomly, provide examination passwords for testees to apply the test through Web as well as collecting and recording the scores data of the tests. Appraising focuses on analyzing the collected/processed data of tests and to generate the statistic report. Zhenming, et. al. (2003) proposed a web-based operational skills examination and evaluation system for computer courses. In another research work by Rashad, et. al (2010), a web-based online examination system was proposed. The system carries out the examination and auto-grading for student's examinations. The system facilitates conducting examinations, collection of answers, auto marking the submissions and production of reports for the test. It supports many kinds of questions. It was used via Internet and is invariably suitable for both local and remote examination. The system could help lecturers, instructors, teachers and others who are willing to create new examinations or edit existing ones as well as students participating in the examinations. The system was built using various open source technologies. AJAX, PHP, HTML and MYSQL database are used in this system. An auto-grading module was generalized to enable different examination and question types. The system was tested in the Mansoura University Quality Assurance Centre. The test proved the validity of using this kind of web based systems for evaluating students in the institutions with high rate of students.

e-Examination (Computer-Based Examination)

In many tertiary institutions in Nigeria, the conduct of examinations as well as the process of producing results has been fraught with various problems leading to inability to release results on time, inability of some students to get their results and several incomplete results. These problems can be mitigated using electronic medium.

E-examination, as used in this paper, refers to a system that involves the conduct of examinations through the web or the intranet using the computer system. Recently, because Internet and database technology have been fully developed, CBT which before was once hosted only on personal computers (PCs) or local area networks (LANs), has now gradually been upgraded to work on the Internet using browsers as the test interface so that users can use it anywhere in the world. WES has been seen to be an effective solution for mass education evaluation (Zhenming et al, 2003).

Computer-based examination and test tools have been applied for different purposes, e.g. placement tests, entry-level tests (prognostic tests), self-assessment tests, regular written and oral examinations (selective and diagnostic examinations), and online surveys.

METHODOLOGY AND SYSTEM DESIGN

This technical paper intends to showcase the development of e-examination application

towards enhancing effective distance education where digital divide is eliminated in access to qualitative education across the globe.

The application was developed using different programming models and languages which include HTML, CSS and PHP (for the front-end interface) and MYSQL (for the backend) and served through a web server, APACHE. The use of HTML and CSS, which is a markup language for information presentation and a styling language respectively, allow for the user-interface to be designed and properly laid out. To enable dynamic content generation, PHP (a web scripting language) is used to generate dynamic contents based on the user of the system and the corresponding content stored in the backend database which is managed by MySQL. The web server is used to serve the webpages to users when they are needed, and also to interpret the PHP scripting commands contained in the page. In other words, the computer simply acts as the medium for students to take examinations, for teachers to construct tests, and for the transmission of test papers.

Web-based Examination System Phases

This WES has three phases. Namely: (i) The presentation phase offers an interface to the user. (ii) The business/logic phase serves as the middleware that is responsible for processing the user's requests. (iii) The database phase or question bank serves as the repository of a pool of questions to be answered by the student.

DEVELOPMENT LANGUAGES

Cascading Style Sheet (CSS)

It is a set of rules that allow user to control how the web document will appear in the web browser. It defines the formatting applied to a Website, including colors, background images, typefaces (fonts), margins, and indentation. The basic purpose of CSS is to allow the designer to define a style (a list of formatting details such as fonts, sizes, and colors) and then, to apply it to one or more portions of HTML pages using a selector. CSS information can be specified in three different places: (i) within the specific tags in the document body (Inline CSS), (ii) at the top of the document within a `<style>` block, or combined with named `<div>` or `` containers in the document body (Embedded CSS), and (iii) in one or more separate files shared across many Web pages (External CSS)

34

Hyper Text Mark-up Language (HTML)

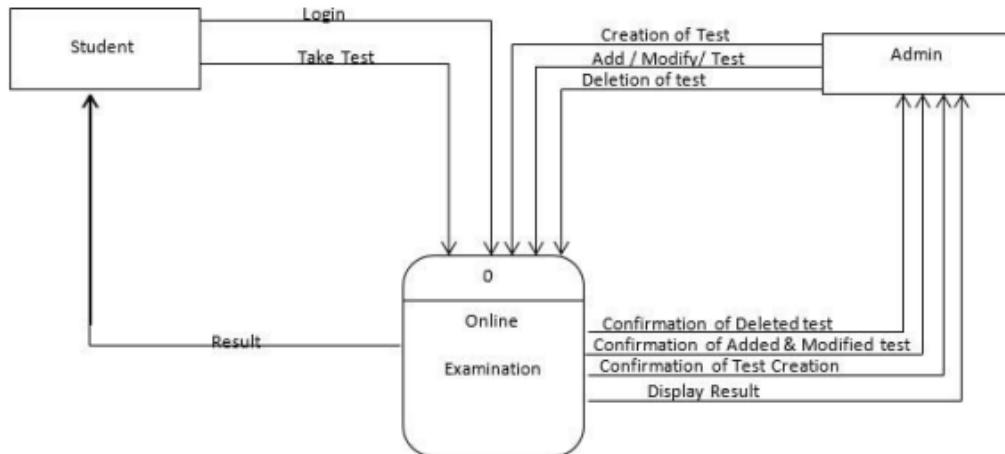
It is the core technology in which all Web pages are written. HTML is not a programming language rather it is a mark-up language for collection of mark-up tags to describe Web pages. Mark-up is made up of tags, and tag names are enclosed in angle brackets.

Hypertext Pre-processor (PHP)

It is a widely-used Open Source general-purpose scripting language that is specifically suited for Web development and can be embedded into HTML. Unlike other CGI script written in other languages like Perl or C, where lots of commands are written to output HTML, the PHP code is enclosed in special start and end tags that allow you to jump into and out of PHP mode. What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server?

Structured Query Language (SQL)

This is the standard language designed to access relational databases.

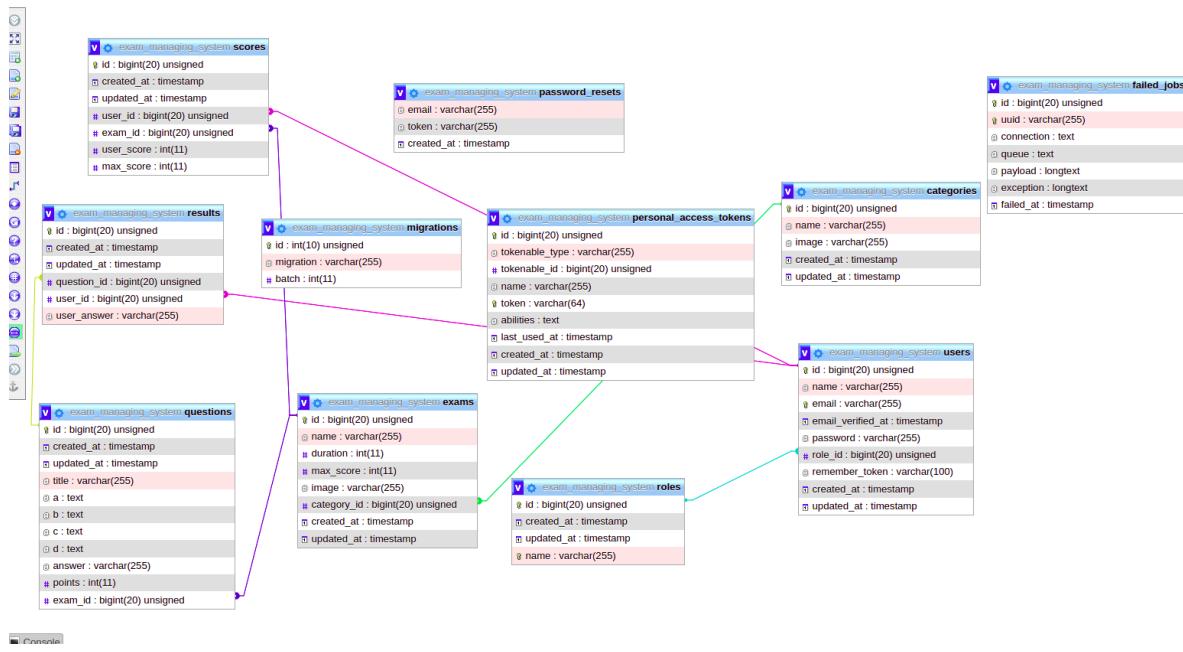


MySQL Workbench

MySQL is the world's most popular open source database, enabling cost effective delivery of reliable, high-performance and scalable web-based and embedded database application. The data in MySQL are stored in tables. MySQL workbench is a unified visual tool for database architectures, developers, and DBAs. It provides data modeling, SQL development and comprehensive administration tools for server configuration, user administration, backup and much more. MySQL workbench enables a DBA, developer, or data architect to manage databases. It includes everything a data modeler needs for creating complex Entity Relational (ER) models, forward and reverse engineering and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort. MySQL workbench delivers visual tools for creating, executing and optimizing SQL queries. The SQL editor provides color syntax highlighting, auto-complete, reuse of SQL snippets and execution history of SQL. The database connections panel enables Figure 1: Architecture of the system 35 developers to easily manage database connections. The object browser provides instant access to database schema and objects.

SYSTEM DESIGN

Database (Backend) Design Database design is concerned with how data is represented and stored within the system. The examination questions, answers, grades, and reviews must be stored in a persistent way. Moreover, we need to keep information about the students. The system stores the above information in a MySQL Database server. Such database has been chosen since it is open source, and there are implementations available for the main architectures



Tables:

Roles Table:

This table specifies the role of the user. In this project we have three roles.

- 1) User: Authorization: the user can access the public site as well as the exam.
- 2) Admin: Authorization: the user can access the public site as well as the entire dashboard. It can't create, edit and delete a user.
- 3) Super Admin: all admin privileges with deleting and editing users

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop ▾ More
2	created_at	timestamp			Yes	NULL			Change Drop ▾ More
3	updated_at	timestamp			Yes	NULL			Change Drop ▾ More
4	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop ▾ More

Users Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	<code>name</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	<code>email</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	<code>email_verified_at</code>	timestamp			Yes	NULL			Change Drop More
5	<code>password</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	<code>role_id</code>	bigint(20)		UNSIGNED	No	None			Change Drop More
7	<code>remember_token</code>	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
8	<code>created_at</code>	timestamp			Yes	NULL			Change Drop More
9	<code>updated_at</code>	timestamp			Yes	NULL			Change Drop More

Categories Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	<code>name</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	<code>image</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	<code>created_at</code>	timestamp			Yes	NULL			Change Drop More
5	<code>updated_at</code>	timestamp			Yes	NULL			Change Drop More

Exams Table:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	<code>name</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	<code>duration</code>	int(11)			No	None			Change Drop More
4	<code>max_score</code>	int(11)			No	None			Change Drop More
5	<code>image</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
6	<code>category_id</code>	bigint(20)		UNSIGNED	No	None			Change Drop More
7	<code>created_at</code>	timestamp			Yes	NULL			Change Drop More
8	<code>updated_at</code>	timestamp			Yes	NULL			Change Drop More

Questions Table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop ▾ More
□	2	created_at	timestamp			Yes	NULL			Change Drop ▾ More
□	3	updated_at	timestamp			Yes	NULL			Change Drop ▾ More
□	4	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop ▾ More
□	5	a	text	utf8mb4_unicode_ci		No	None			Change Drop ▾ More
□	6	b	text	utf8mb4_unicode_ci		No	None			Change Drop ▾ More
□	7	c	text	utf8mb4_unicode_ci		No	None			Change Drop ▾ More
□	8	d	text	utf8mb4_unicode_ci		No	None			Change Drop ▾ More
□	9	answer	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop ▾ More
□	10	points	int(11)			No	None			Change Drop ▾ More
□	11	exam_id	bigint(20)		UNSIGNED	No	None			Change Drop ▾ More

Results Table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop ▾ More
□	2	created_at	timestamp			Yes	NULL			Change Drop ▾ More
□	3	updated_at	timestamp			Yes	NULL			Change Drop ▾ More
□	4	question_id	bigint(20)		UNSIGNED	No	None			Change Drop ▾ More
□	5	user_id	bigint(20)		UNSIGNED	No	None			Change Drop ▾ More
□	6	user_answer	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop ▾ More

Scores Table:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop ▾ More
□	2	created_at	timestamp			Yes	NULL			Change Drop ▾ More
□	3	updated_at	timestamp			Yes	NULL			Change Drop ▾ More
□	4	user_id	bigint(20)		UNSIGNED	No	None			Change Drop ▾ More
□	5	exam_id	bigint(20)		UNSIGNED	No	None			Change Drop ▾ More
□	6	user_score	int(11)			No	None			Change Drop ▾ More
□	7	max_score	int(11)			No	None			Change Drop ▾ More

Backend Design:

The backend uses laravel which implements MVC architecture.

Introduction

The default Laravel application structure is intended to provide a great starting point for both large and small applications. But you are free to organize your application however you like. Laravel imposes almost no restrictions on where any given class is located - as long as Composer can autoload the class.

The Root Directory

The App Directory

The app directory contains the core code of your application. We'll explore this directory in more detail soon; however, almost all of the classes in your application will be in this directory.

The Bootstrap Directory

The bootstrap directory contains the app.php file which bootstraps the framework. This directory also houses a cache directory which contains framework generated files for performance optimization such as the route and services cache files. You should not typically need to modify any files within this directory.

The Config Directory

The config directory, as the name implies, contains all of your application's configuration files. It's a great idea to read through all of these files and familiarize yourself with all of the options available to you.

The Database Directory

The database directory contains your database migrations, model factories, and seeds. If you wish, you may also use this directory to hold an SQLite database.

The Lang Directory

The lang directory houses all of your application's language files.

The Public Directory

The public directory contains the index.php file, which is the entry point for all requests entering your application and configures autoloading. This directory also houses your assets such as images, JavaScript, and CSS.

The Resources Directory

The resources directory contains your views as well as your raw, un-compiled assets such as CSS or JavaScript.

The Routes Directory

The routes directory contains all of the route definitions for your application. By default, several route files are included with Laravel: web.php, api.php, console.php, and channels.php.

The web.php file contains routes that the RouteServiceProvider places in the web middleware group, which provides session state, CSRF protection, and cookie encryption. If your application does not offer a stateless, RESTful API then it is likely that all of your routes will most likely be defined in the web.php file.

The api.php file contains routes that the RouteServiceProvider places in the api middleware group. These routes are intended to be stateless, so requests entering the application through these routes are intended to be authenticated via tokens and will not have access to session state.

The console.php file is where you may define all of your closure based console commands. Each closure is bound to a command instance allowing a simple approach to interacting with each command's IO methods. Even though this file does not define HTTP routes, it defines console based entry points (routes) into your application.

The channels.php file is where you may register all of the event broadcasting channels that your application supports.

The Storage Directory

The storage directory contains your logs, compiled Blade templates, file based sessions, file caches, and other files generated by the framework. This directory is segregated into app, framework, and logs directories. The app directory may be used to store any files generated by your application. The framework directory is used to store framework generated files and caches. Finally, the logs directory contains your application's log files.

The storage/app/public directory may be used to store user-generated files, such as profile avatars, that should be publicly accessible. You should create a symbolic link at public/storage which points to this directory. You may create the link using the php artisan storage:link Artisan command.

The Tests Directory

The tests directory contains your automated tests. Example PHPUnit unit tests and feature tests are provided out of the box. Each test class should be suffixed with the word Test. You may run your tests using the phpunit or php vendor/bin/phpunit commands. Or, if you would like a more detailed and beautiful representation of your test results, you may run your tests using the php artisan test Artisan command.

The Vendor Directory

The vendor directory contains your Composer dependencies.

The App Directory

The majority of your application is housed in the app directory. By default, this directory is namespaced under App and is autoloaded by Composer using the PSR-4 autoloading standard.

The app directory contains a variety of additional directories such as Console, Http, and Providers. Think of the Console and Http directories as providing an API into the core of your application. The HTTP protocol and CLI are both mechanisms to interact with your application, but do not actually contain application logic. In other words, they are two ways of issuing commands to your application. The Console directory contains all of your Artisan commands, while the Http directory contains your controllers, middleware, and requests.

A variety of other directories will be generated inside the app directory as you use the make Artisan commands to generate classes. So, for example, the app/Jobs directory will not exist until you execute the make:job Artisan command to generate a job class.

Many of the classes in the app directory can be generated by Artisan via commands. To review the available commands, run the php artisan list make command in your terminal.

The Broadcasting Directory

The Broadcasting directory contains all of the broadcast channel classes for your application. These classes are generated using the make:channel command. This directory does not exist by default, but will be created for you when you create your first channel. To learn more about channels, check out the documentation on event broadcasting.

The Console Directory

The Console directory contains all of the custom Artisan commands for your application. These commands may be generated using the make:command command. This directory also houses your console kernel, which is where your custom Artisan commands are registered and your scheduled tasks are defined.

The Events Directory

This directory does not exist by default, but will be created for you by the event:generate and make:event Artisan commands. The Events directory houses event classes. Events may be used to alert other parts of your application that a given action has occurred, providing a great deal of flexibility and decoupling.

The Exceptions Directory

The Exceptions directory contains your application's exception handler and is also a good place to place any exceptions thrown by your application. If you would like to customize how your exceptions are logged or rendered, you should modify the Handler class in this directory.

The Http Directory

The Http directory contains your controllers, middleware, and form requests. Almost all of the logic to handle requests entering your application will be placed in this directory.

The Jobs Directory

This directory does not exist by default, but will be created for you if you execute the make:job Artisan command. The Jobs directory houses the queueable jobs for your

application. Jobs may be queued by your application or run synchronously within the current request lifecycle. Jobs that run synchronously during the current request are sometimes referred to as "commands" since they are an implementation of the command pattern.

The Listeners Directory

This directory does not exist by default, but will be created for you if you execute the `event:generate` or `make:listener` Artisan commands. The Listeners directory contains the classes that handle your events. Event listeners receive an event instance and perform logic in response to the event being fired. For example, a `UserRegistered` event might be handled by a `SendWelcomeEmail` listener.

The Mail Directory

This directory does not exist by default, but will be created for you if you execute the `make:mail` Artisan command. The Mail directory contains all of your classes that represent emails sent by your application. Mail objects allow you to encapsulate all of the logic of building an email in a single, simple class that may be sent using the `Mail::send` method.

The Models Directory

The Models directory contains all of your Eloquent model classes. The Eloquent ORM included with Laravel provides a beautiful, simple ActiveRecord implementation for working with your database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow you to query for data in your tables, as well as insert new records into the table.

The Notifications Directory

This directory does not exist by default, but will be created for you if you execute the `make:notification` Artisan command. The Notifications directory contains all of the "transactional" notifications that are sent by your application, such as simple notifications about events that happen within your application. Laravel's notification feature abstracts sending notifications over a variety of drivers such as email, Slack, SMS, or stored in a database.

The Policies Directory

This directory does not exist by default, but will be created for you if you execute the `make:policy` Artisan command. The Policies directory contains the authorization policy classes for your application. Policies are used to determine if a user can perform a given action against a resource.

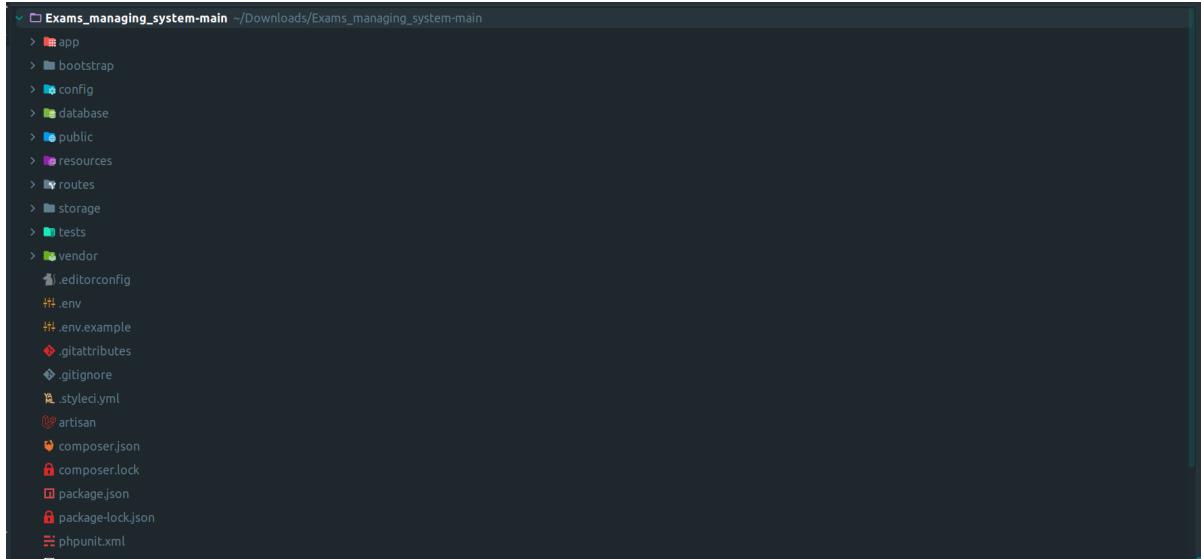
The Providers Directory

The Providers directory contains all of the service providers for your application. Service providers bootstrap your application by binding services in the service container, registering events, or performing any other tasks to prepare your application for incoming requests.

In a fresh Laravel application, this directory will already contain several providers. You are free to add your own providers to this directory as needed.

The Rules Directory

This directory does not exist by default, but will be created for you if you execute the make:rule Artisan command. The Rules directory contains the custom validation rule objects for your application. Rules are used to encapsulate complicated validation logic in a simple object. For more information, check out the validation documentation.



```
Exams_managing_system-main
├── app
├── bootstrap
├── config
├── database
├── public
├── resources
└── routes
├── storage
└── tests
└── vendor
  ├── .editorconfig
  ├── .env
  ├── .env.example
  ├── .gitattributes
  ├── .gitignore
  ├── .styleci.yml
  ├── artisan
  ├── composer.json
  ├── composer.lock
  ├── package.json
  └── package-lock.json
└── phpunit.xml
```

Routes:

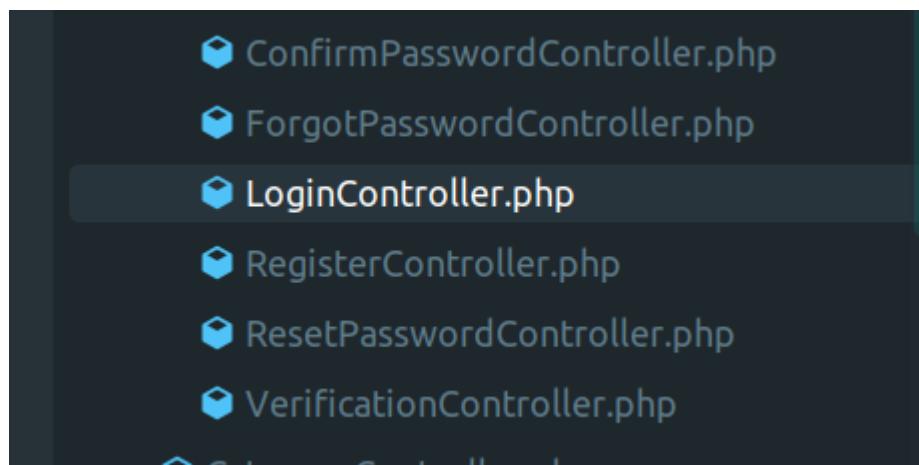
Admin Routes:

```
//admin routes
Route::prefix('admin')->name('admin.')->middleware('admin.auth')->group(function () {
    Route::get('/', function () {
        return view('admin.index2');
    });
    Route::resource('users', UserController::class)->middleware('super_admin.auth');
    Route::resource('categories', CategoryController::class);
    Route::resource('exams', ExamController::class);
    Route::resource('questions', QuestionController::class);
    Route::resource('results', ResultController::class);
    Route::resource('scores', ScoreController::class);
});;
```

public routes:

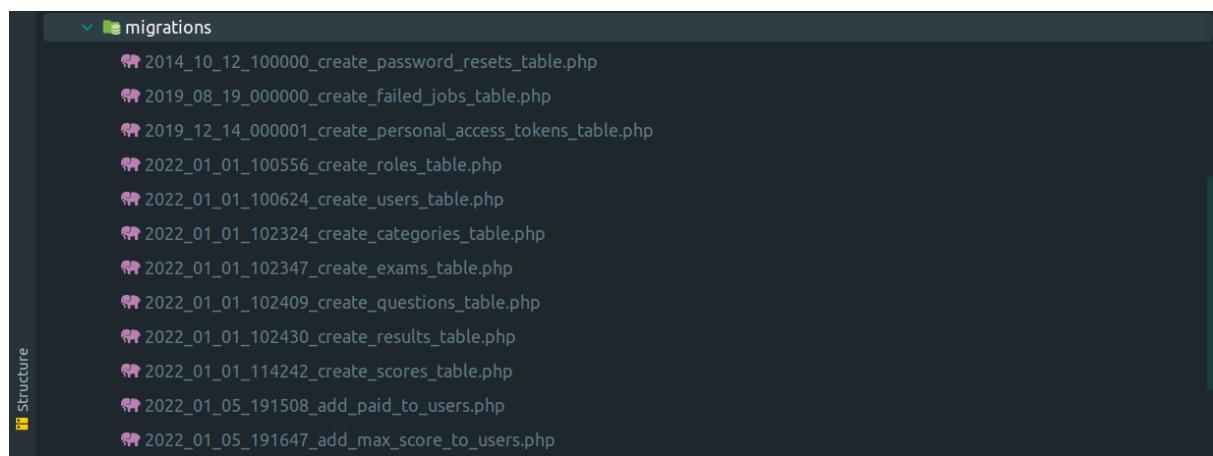
```
//display results
Route::get( uri: '/results/{id}', [ResultController::class, 'show']);
Auth::routes();
Route::get( uri: '/', [HomeController::class, 'index'])->name( name: 'home');
Route::get( uri: '/home', function () {
    return view( view: 'pages.index');
})->name( name: 'main_page');
Route::get( uri: '/exam/{id}', [ExamController::class, 'show'])->name( name: 'exam-single')->middleware( middleware: 'auth');
Route::get( uri: '/Exams', [ExamController::class, 'showAllExams'])->name( name: 'Exams')->middleware( middleware: 'auth');
Route::get( uri: '/profile', [ScoreController::class, 'showAllUserScores'])->name( name: 'profile')->middleware( middleware: 'auth');
Route::post( uri: '/api/', [ResultController::class, 'set_results']);
```

Controllers:



Auth controllers: responsible for the authentication of the user

Migrations



Login function:

```
    */
    public function login(Request $request)
    {
        $this->validateLogin($request);

        // If the class is using the ThrottlesLogins trait, we can automatically throttle
        // the login attempts for this application. We'll key this by the username and
        // the IP address of the client making these requests into this application.
        if (method_exists($this, 'hasTooManyLoginAttempts') &&
            $this->hasTooManyLoginAttempts($request)) {
            $this->fireLockoutEvent($request);

            return $this->sendLockoutResponse($request);
        }

        if ($this->attemptLogin($request)) {
            if ($request->hasSession()) {
                $request->session()->put('auth.password_confirmed_at', time());
            }

            return $this->sendLoginResponse($request);
        }
    }
}
```

```
* @throws Illuminate\Validation\ValidationException
*/
protected function validateLogin(Request $request)
{
    $request->validate([
        $this->username() => 'required|string',
        'password' => 'required|string',
    ]);
}

/**
 * Attempt to log the user into the application.
*/
```

Logout function:

```
public function logout(Request $request)
{
    $this->guard()->logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    if ($response = $this->loggedOut($request)) {
        return $response;
    }

    return $request->wantsJson()
        ? new JsonResponse([], status: 204)
        : redirect(to: '/');
}
```

Project Controllers:

```
📦 CategoryController.php  
📦 Controller.php  
📦 ExamController.php  
📦 HomeController.php  
📦 QuestionController.php  
📦 ResultController.php  
📦 ScoreController.php  
📦 UserController.php
```

Middlewares:

The system authenticate user roles using middlewares

```
🔒 Admin.php  
📦 Authenticate.php  
📦 EncryptCookies.php  
📦 PreventRequestsDuringMaintenance.p  
📦 RedirectIfAuthenticated.php  
📦 SuperAdmin.php  
📦 TrimStrings.php  
📦 TrustHosts.php  
📦 TrustProxies.php  
📦 VerifyCsrfToken.php
```

The main middleware here is the admin middleware:

```
class Admin
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse) $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        $name = Auth::user()->role->name ?? null;
        if ($name === "admin" || $name === "super_admin") {
            return $next($request);
        }
        abort( code: 403);
    }
}
```

Handles superuser and user.

User Interface (Frontend) Design

Usability is the ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system or component. This usability of a system is made less more or less stressful by the usability and complexity of the user interface. The user interface of a system is therefore the part of the system that the end user interacts with. User interface design is concerned with how users add information to the system and with how the system presents information back to them.

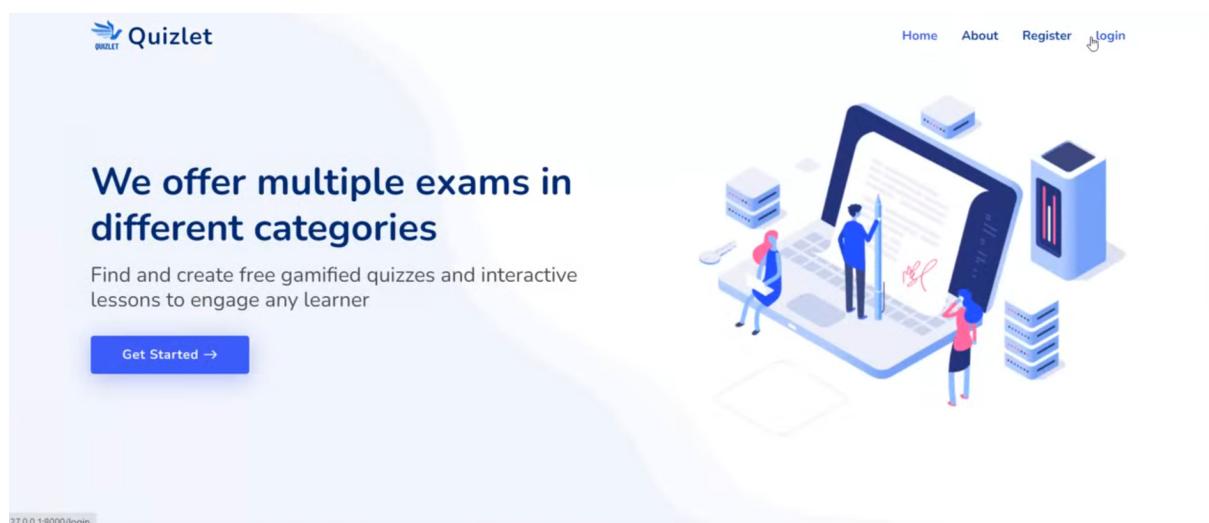
Bootstrap

Bootstrap is a collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components as well as optional JavaScript extensions. Bootstrap was developed at

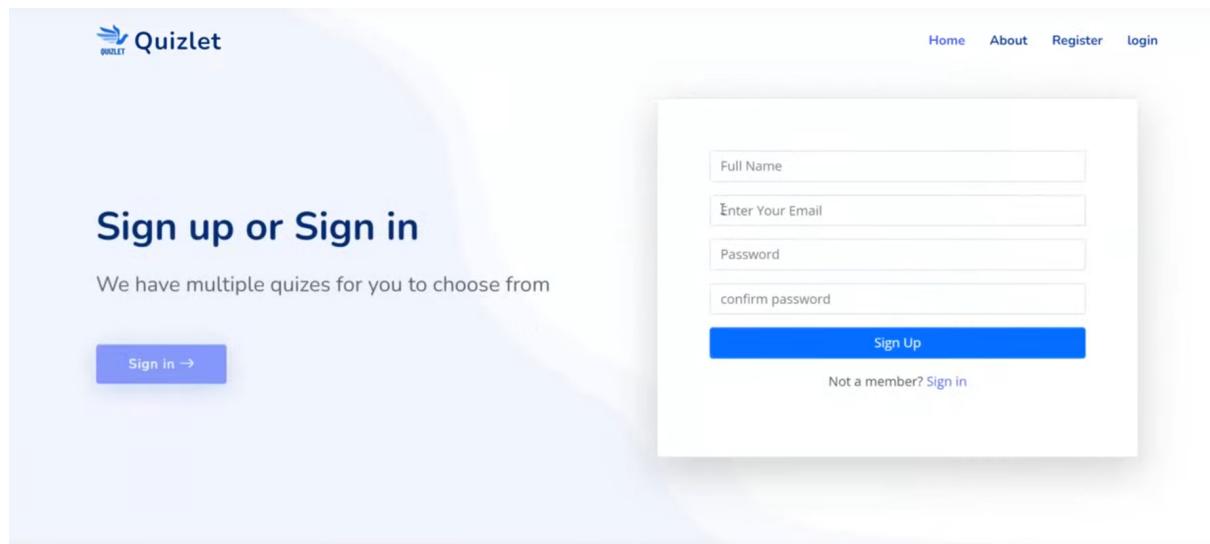
Twitter as a framework to encourage consistency across internal tools. Bootstrap is compatible with all major browsers and it also supports responsive design i.e. the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (PC, tablet, mobile phone). Bootstrap works by proving a clean and uniform solution to the most common, everyday interface tasks developers come across. It is flexible enough to work for many unique design needs.

User Journey:

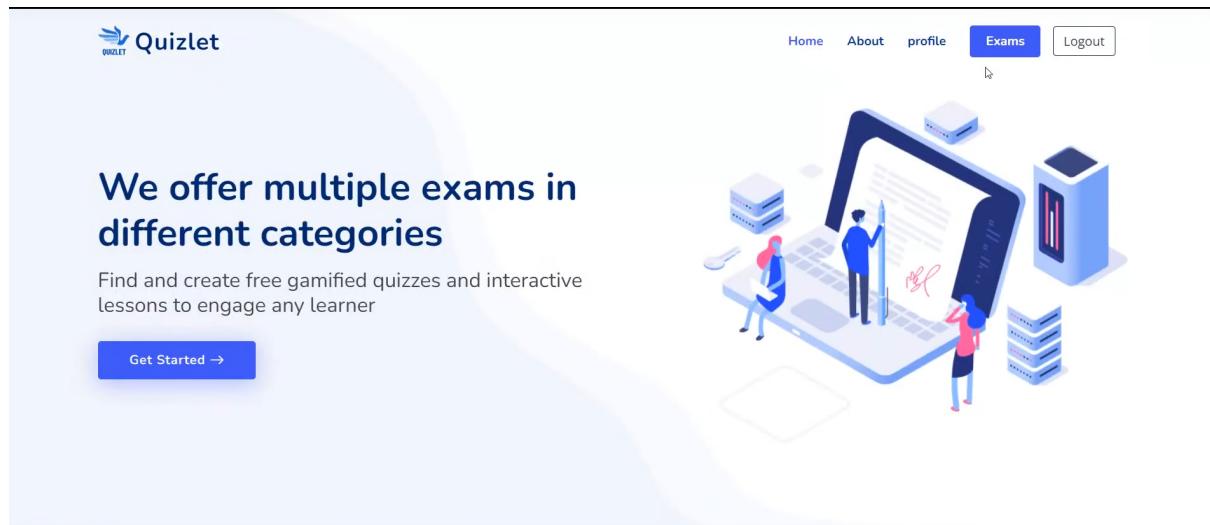
The user enters the landing page:



If he has an exam he will register



After Registration the user is logged in:



Notice that the user can choose an exam.

EXAMS

Check our latest exams



After that the exam details will be displayed



[Home](#) [About](#) [profile](#) [Exams](#) [Logout](#)

HTML Exam



Final Score is **50 point**

The exam is from web-development category and you have 45 minutes to finish it please invest your time wisely



[Start now](#)

The exam will start:

Questions

What does a markup language use to identify content?

1/10

Time Left

29

functions

scripts

tags

commands

next

After the user finishes the exam

Your result is 7.5/25

wrong
Answercorrect
Answer

What does a markup language use to identify content?

- a functions
- b scripts
- c tags

What is HTML?

- a Hypertext Library
- b Programming Language
- c Markup Language

When formatting text, can you get the same result when using different tags?

- a probably
- b yes
- c sometimes

All the answers will be displayed

What does a markup language use to identify content?

- a functions
- b scripts
- c tags**
- d commands

What is HTML?

Unanswered

- a Hypertext Library
- b Programming Language
- c Markup Language**
- d Scripting Language

When formatting text, can you get the same result when using different tags?

- a probably
- b yes
- c sometimes**
- d no

User Profile:

Welcome Awni

Here you can see all your exams that you have taken and their scores

Created at	Exam	exam_category	score	exam_max_score	exam_duration	show results
2022-01-09 10:10:26	html	web-development	15	50	45	show answers
2022-01-09 10:10:15	html	web-development	15	50	45	show answers
2022-01-09 10:09:44	html	web-development	15	50	45	show answers

Admin Dashboard:


Search for datas & reports...
Awni

You are here: Home / Dashboard

- [Dashboard](#)
- [Categories](#)
- [Exams](#)
- [questions](#)
- [results](#)
- [scores](#)
- [Home](#)

Welcome Awni

5

MEMBERS ONLINE



4

EXAMS WE HAVE



116

TOTAL EXAMS TAKEN





In the dashboard the admin can see all of the statistics of the website

QuizLogic:

The Quiz was built using JS in this section I will Discuss The logic of the quiz:

The architecture of the code is functional programing so I will first discuss functional programing.

Functions "Encapsulate" a task (they combine many instructions into a single line of code). Most programming languages provide many built in functions that would otherwise require many steps to accomplish, for example computing the square root of a number. In general, we don't care how a function does what it does, only that it "does it"!

When a function is "called" the program "leaves" the current section of code and begins to execute the first line inside the function. Thus the function "flow of control" is:

1. The program comes to a line of code containing a "function call".
 2. The program enters the function (starts at the first line in the function code).
 3. All instructions inside of the function are executed from top to bottom.
 4. The program leaves the function and goes back to where it started from.
 5. Any data computed and RETURNED by the function is used in place of the function in the original line of code.
-

Why do we Write Functions?

1. They allow us to conceive of our program as a bunch of sub-steps. (Each sub-step can be its own function. When any program seems too hard, just break the overall program into sub-steps!)
2. They allow us to reuse code instead of rewriting it.
3. Functions allow us to keep our variable namespace clean (local variables only "live" as long as the function does). In other words, function_1 can use a variable called i, and function_2 can also use a variable called i and there is no confusion. Each variable i only exists when the computer is executing the given function.
4. Functions allow us to test small parts of our program in isolation from the rest. This is especially true in interpreted languages, such as Matlab, but can be useful in C, Java, ActionScript, etc.

Code:

```

const generateQuestionMarkup=function (question){
  const container = document.querySelector( selectors: "#questionContainer");
  container.innerHTML=""

  const markup=`<section class="questions-section mt-5 mb-5" style="height:85vh !important;" >
    <a href="/" class="d-block h6 ms-auto home-btn"><i class="fas fa-home"></i></a>
    <div class="container">
      <h2 class="section-title">Questions</h2>
      <div class="row">
        <div class="question d-flex">
          <h2 class="question">${question.title} </h2>

          <h4 class="question-count">${counter}/10</h4>

        </div>
        <div class="counter_wrap">
          <div class="timer_wrap">
            Time Left
            <div class="timer">30</div>
          </div>
        </div>
      </div>
    </div>
  
```

generateQuestionMarkup:
 generates a styles markup of the quiz
storeCounter
 saves the questions count
inputCheck
 checks if the input is selected

CONCLUSION AND RECOMMENDATIONS

It is not enough to focus on the passing required skills to the learners in distance education but to equally furnish them with their performances shortly after evaluation without hitch. Consequently, the developed WES is capable of solving the associated problems with the traditional test methods and equally promotes distance education. Figure 8. Result Page Figure 9. Database View Page 40 When online method of instruction is used to acquire skills in higher education, the application can be used for efficient assessments regardless of the location of the examinees across the globe. It is possible with this system to space the period of examination without compromising quality and integrity of the examination. The system has the potentials to reduce drastically examination malpractice as applicants are duly authenticated online, real-time before taking the examination and the integrity of the result could also be enhanced since the candidates have access to instant result checking. Hence, in the current era of distance education prompted by the adoption of ICT, the eExamination has the advantage of being easy to administer, ability to offer applicants instant results, easy verification, devoid of paper work and long-time involved in marking examination scripts which in most cases are prone to errors and misplacement of some scripts due to the large volume of scripts that has to be marked and accessed. The system also saves the instructors from sufferings and boring grading of works as well as examinees' access to results thereby promoting efficient distance education system. If the e-examination system is fully optimized and it is introduced into the institutions, it will go a long way to control and check examination malpractices and all fraudulent acts associated with the manual process of writing examination. However, for the system to be adopted on a large scale, efforts should be intensified to ascertain its disadvantage on accounts of IT illiteracy on the part of the students', by making the interface easy to interact with. Also, to ensure that the e-examination system is not intensified by those that may want to engage in any form of examination malpractice, the addition of user authenticated

protocol/methods such as biometrics (fingerprint, retina, iris etc) identification will be of good help. In future, we intend to address the limitations of the current application by incorporating online collation of results from various courses examined, compute Cumulative Grade Point Aggregate (CGPA) and generation of transcripts (with necessary security features) to foster implementation of distance education.