**REAL TIME TEST MONITORING**

A project work submitted to the Jamal Mohamed College (Autonomous), Tiruchirappalli in partial fulfillment of the requirements for the award of the degree of

**MASTER OF COMPUTER APPLICATIONS**

Submitted

by

**R. SAFDHAR**

**(Reg.No.18MCA005)**

Under the Guidance of

**DR. O. A. MOHAMED JAFAR M.Phil., Ph.D.,**

**Associate Professor**



**PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE**

**JAMAL MOHAMED COLLEGE (AUTONOMOUS)**

**College with Potential for Excellence**

**Accredited (3rd Cycle) with ‘A’ Grade by NAAC**

**(Affiliated to Bharathidasan University)**

**TIRUCHIRAPPALLI – 620 020**

**APRIL 2021**

**JAMAL MOHAMED COLLEGE (AUTONOMOUS)**

**College with Potential for Excellence**

**Accredited (3rd Cycle) with ‘A’ Grade by NAAC**

**(Affiliated to Bharathidasan University)**

**TIRUCHIRAPPALLI – 620 020**



This is to certify that this project work entitled

**REAL TIME TEST MONITORING**

is a bonafide record of the project work done

by

**R. SAFDHAR**

**(Register Number: 18MCA005)**

at Jamal Mohamed College (Autonomous), Tiruchirappalli during the year 2017-2020 in partial fulfillment of the requirements for the award of the degree of

**MASTER OF COMPUTER APPLICATIONS**

**DR. O. A. MOHAMED JAFAR Dr. G. RAVI**

**Associate Professor Head of the Department**

Submitted for the Viva Voce examination held at Jamal Mohamed College(Autonomous), Tiruchirappalli – 20 on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Internal Examiner External Examiner**

**Dr. O. A. MOHAMED JAFAR M.Phil. ,Ph.D.,**

**Associate Professor,**

**Dept. of Computer Science,**

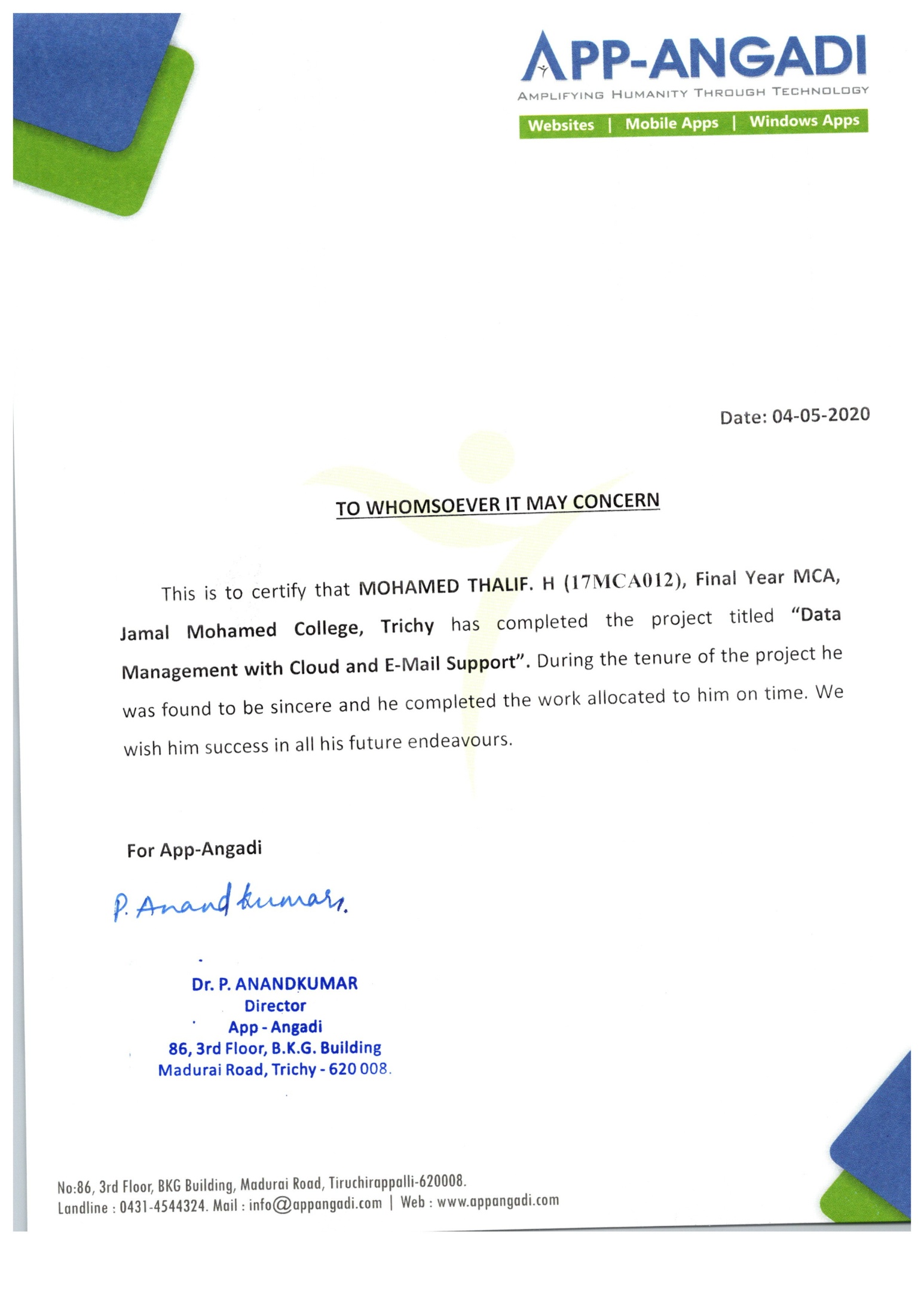
**Jamal Mohamed College (Autonomous),**

**Tiruchirappalli-620 020.**

**CERTIFICATE**

This is to certify that this project entitled “**REAL TIME TEST MONITORING**” submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications to the Jamal Mohamed College, Tiruchirappalli is a bonafide record of the work done by **R. SAFDHAR (Reg.No.18MCA005)** under my supervision and guidance**.**

**Date:**  **Signature of the Guide**



**ACKNOWLEDGEMENT**

First and foremost I am grateful to the Almighty God who has showered his blessings on me all through my life and throughout this project.

Words are inadequate to express our heartiest, sincere and grateful thanks to the following persons.

Our first teachers and well-wishers, our first guide, we first express our deepest gratitude to our parents who were our strength and for their constant support in our every single step.

I place on record my sincere gratitude from the bottom of my heart to our beloved Principal **Hajee Dr. S.ISMAIL MOHIDEEN, M.Sc., M.Phil., PGDCA., Ph.D. Principal,** Jamal Mohamed College(Autonomous) for giving me an opportunity to do this course in this prestigious institution.

I am deeply indebted to **Dr. G. Ravi, M.C.A., PGDBA., M.Phil., Ph.D. – Head, Department of Computer Science** and **Dr. D.I. George Amalarethinam, M.C.A., M.Phil., PGDBA., Ph.D. – Bursar & Director of MCA**, Jamal Mohamed College(Autonomous) for granting permission for this project work.

With the deep appreciation and sincere gratitude, I humbly and deeply thankful to my project guide **Dr. O.A. Mohamed Jafar**. It is an immense pleasure to show my gratitude to him for he has been the greatest support in my project work.

Finally, I would like to express my love and gratitude to my beloved parents to who I dedicate this project for their love, support and encouragement whenever I feel disdained.

**R. SAFDHAR**

**(18MCA005)**

## **ABSTRACT**

This project title is “**Real Time Test Monitoring”**is a Desktop Application developed to monitor the Real Time Test. The purpose of test monitoring is to give feedback and visibility about test activities. Information to be monitored and collected automatically and used to measure exit criteria. The computerized system will improve the existing system that currently processes it manually. The computerized system will improve the existing system to provide the right information at the right time and will show the Upcoming Test Details of Student and Ongoing Progress of Test. This will help to get the report of Students if they are completed. It monitors the behaviour of Students while they write tests. The System Administrator and the faculties can get details anytime without any delay.

The features of the system include the following aspects,

* Backing up the files either automatically or manually
* Restoring the data whenever it is needed
* Track the Test Status

These systems are real-time dashboards and visualized data are updated continuously every 5 seconds the action occurs. It will help the Management to monitor the status of ongoing Tests. No worry about Test Report, it is stored in the database automatically whenever they want, they can generate Report. While Monitoring Ongoing Test it also monitors the login status of student and progress of the test.

|  |  |  |
| --- | --- | --- |
| **CHAPTER NUMBER** | **INDEX** | **PAGE NUMBER** |
| 1 | **INTRODUCTION**   * 1. About the Project   2. Company Profile | 1  2 |
| 2 | **PROBLEM DEFINITION AND DESCRIPTION** |  |
|  | **SYSTEM STUDY** |  |
| 3 | * 1. Existing System   2. Disadvantage of Existing System | 5  5 |
|  | 3.3 Proposed System | 5 |
|  | 3.4 Advantage of Proposed System | 6 |
|  | **SYSTEM ANALYSIS** |  |
| 4 | * 1. Package Selected   2. Hardware Requirements   3. Software Requirements   4. Dataflow Diagram |  |
| 5 | **SYSTEM DESIGN**  5.1 Architectural Design  5.2 Database Diagram  5.3 Data Dictionary |  |
| 6 | **SYSTEM DEVELOPMENT**  6.1 Special Features of Language |  |
| 7 | **IMPLEMENTATION AND TESTING** |  |
| 8 | **CONCLUSION** |  |
|  | **BIBLIOGRAPHY** |  |
| **ANNEXURE** | **A SCREEN INTERFACES** |  |

**CONTENTS**

**CHAPTER 1**

**INTRODUCTION**

## **About the Project**

“**Real Time Test Monitoring”** is a Test Monitoring Application. This monitoring software is employed to monitor the test takers and establish the exam’s credibility and authenticity. It is overseeing an exam and monitoring the students. In case a student has an issue while writing at the time of the exam, from our system the report says what issue they had when they write the exam. This System Shows the Upcoming Schedule with date and time, it show the Number of Batches write the ongoing test and Students overall present and it also show individual batch Students present and we can get report of completed tests.

These all data visualize in the form representation in the dashboard. It helps to see the these all data at one page. This dashboard is a real-time dashboard allowing to monitor activity as it happens throughout the whole process from upcoming to completed. Visualized data are updated continuously in less than a second after the action occurs. It will help to get enhanced report based on the visualized data. This help to find the how many batches are completed, how many students are present , how many batches are having test in feature, how they are completed their test.

* Shows details of batches who have tests on upcoming days
* At the day of test it shows, is all batches are taken exam or not
* Is all batches students are present or not
* Are they online or offline. If they are not online, their details will be shown in offline page
* Shows the Number of students presented at the particular batch
* It can give report of batches who completed test

These all data details will help the administrator to find the students and batch how they are performing.

This dashboard project is developed for desktop application by using WPF, C#, MS SQL Server and RDLC.

**CHAPTER 1**

## **1.2 Company Profile:**

Company Name : **APP – ANGADI**

External Guide Name : **Dr. P. Anand Kumar MCA, PhD.,**

Address : II Floor, BKG Building, No. 86 Madurai road, Tiruchirappalli– 620 008

Email : [root.anand@gmail.com](mailto:root.anand@gmail.com)

Website : [www.appangadi.com](http://www.appangadi.com)

#### **About Company**



App – Angadi is a startup established in December 2009 as a private company to develop User Centered Applications. It aims at delivering applications on all platforms such as iOS, Android and Windows at time with full support and unlimited revisions. At App - Angadi, we combine business and technical knowledge based on the requirements of the client and ensure maximum Customer Satisfaction.

Our main aim is to provide a high quality application that complies to the given specification and deliver it within the stipulated time frame. We also support development lifecycle by providing functional and technical requirements through coding and testing. We are ready to deploy our extensive resources to meet customer’s challenges. It also acts as a great learning platform for students who joins here as an intern to have a real time experience on a working environment.

**CHAPTER 2**

# **PROBLEM DEFINITION AND DESCRIPTION**

The “**Real Time Test Monitoring”** is a Test Monitoring System for Students. If Any Batch has Exam it shows the Upcoming Schedule. If any batch writes a test it shows the details of batch how many students attend the test now, how many students completed and how many students are offline. It shows how many batches completed the test.

The modules of this project are categorized based on Management needs and the description modules are mentioned as follows:

**Real Time Test Monitoring Dashboard:**

Real Time Test Monitoring Dashboard will visualize the following,

* Upcoming
* Today
* Completed

**Upcoming:**

In Upcoming, it shows the upcoming tests of a batch. It shows the test date, time and Start Type which is manual or auto-launch. Here we can change the manual into auto or auto into the manual. If any test set into autostart it launch automatically at the time for that particular batch. They can take the test at that time and date. If any test set into manual, that particular test will be move exactly at that time on that date onto Today Module to launch. And these all stored in the Database after every change.

**Today:**

In Today, it shows the Ongoing Tests. If more than one batch writes the test, it shows all students details of the ongoing test. It shows How many Tests are going at that time, How Many Students have Attended the Test, How many Students have Attended the Test but currently offline, How many students are completed and the Remaining Duration of the Test.

Each Student details have Student Id, Name of the Student, Class, Batch name, How many questions Attended, Test Completed or Not and How They are Submit their Test like They Complete Between Duration of Test or it’s a Timeout or Tab Closed.

**CHAPTER 2**

**Completed:**

In Completed, it shows the details of the Completed Test. It helps to get a report of the particular batch test, we can get a subject report.

The report have Test Name, Number of Questions , Name of the student, Number of Correct Answer, Number Wrong Answer, Mark and Average.

We get a report with the help of RDLC. RDLC stands for Report Definition Language Client-Side using Microsoft Reporting Technology. It is not a third-party report and is a built-in Microsoft Visual Studio.

If the Correct Answer Option is accidentally changed by any mistake and want to change that Question Answer we can change it by click on Edit and Change Option of Correct Answer for that Question, After that change, it will update all students who write that as correct answer.

**CHAPTER 3**

# **SYSTEM STUDY**

## **3.1 Existing System**

In the present System there is no possibility monitor multiple batches who write the test at the same time and get the report of that student after completion. The traditional exam invigilation is where an invigilator conducts exam at physical testing centers. There is a fixed date, time and venue for the exam where in the invigilator and the applicants are face to face during the supervision. And traditional invigilation requires a ratio of one invigilator for approximately 36 candidates.

We want to assign one person to Mark the attendance and authenticate students appearing for the exam. Monitor applicants during the examination. And Taken time to give report of the particular students mark. These all take time too long

**3.2 Disadvantages of Existing System**

* Difficult to monitor all student of all batches at one time
* It’s a time consuming process
* If may it happened in manual process so sometime the data may not be the correct one. May some important information be lost or overloaded by humans.
* Slow process to give report of test and sometimes may it takes too long for report

## **Proposed System**

The Proposed System is a web-based dashboard and It helps management to see the Test Monitoring on One Page. The dashboards are real-time dashboards that allow to monitor the students who write test on their schedule like their login time, their test duration, attendance to that student, giving report to batch after completing the test.

If student is not present it is easier to inform management to who is not still not came for exam, if student is offline we can inform that student to came online or resolve their issue. These all not taken too long to find. It shows automatically from the database.

**CHAPTER 3**

**3.4 Advantages of Proposed System**

* Faster and Effective access the date
* Less Time-Consuming
* Real Time dashboard
* User-friendly
* Easily understandable data visualization
* No need to data entry it is automated from the database

**CHAPTER 4**

# **SYSTEM CONFIGURATION**

## **Packages Selected:**

Front End : .Net Framework 4.8, WPF(XAML),C#

Back End : MS SQL SERVER

## **Hardware Requirements:**

Processor : Intel core i3

Processor Speed : 2.53GHz

Main Storage : 4 GB RAM

Hard Disk Capacity : 80 GB

## **Software Requirements:**

Operating System : Windows 10

Package : .NET SDK

Web Server : Internet Information Server – IIS

IDE : Microsoft Visual Studio 2019

Database : Microsoft SQL Server 2019

Documentation Tool : Microsoft Word 2019

## **4.4 Data Flow Diagram**

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model.

**CHAPTER 4**

**DFD Symbols**

1. **Rectangle**

Source or Destination for data, external entity

1. **Arrow**

Flow of Data

1. **Circle**

Process of data

1. **Open Rectangle**

Data Store



1. A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data in the beginning process.
4. A data flow to a data store means update (delete or change).

**CHAPTER 4**

1. A data flow from a data store means retrieve or use.

A Data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package. In this fig 3.3.1 shows how the Dashboard get the details and flow diagram.

Upcoming Dashboard get the data from student database, it get details of upcoming batch test date details, test name, test launch type.

Ongoing page get the data from student database, it get details of Ongoing test details student name, batch name, remaining duration, attendance of students, offline details, if test is manual launch then it get that details also.

Completed page get the data from student database, it get details of Completed test details of batches. Get the test name, date of test, number of questions, which class , answer keys to edit if any changes want to update in answer key. If any change made on answer key it will be update on student database.

Student Database

UPCOMING DASHBOARD

ONGOING DASHBOARD

COMPLETED DASHBOARD

**Fig 4.4.1:** Data Flow Diagram for Real Time Test Monitoring

**CHAPTER 5**

# **DESIGN AND DEVELOPMENT**

# **5.1 Architectural Design:**

SYSTEM ADMIN

COLLECTED DATA

DATA BASE

REAL TIME TEST MONITORING

COMPLETED TEST

UPCOMING TEST

ONGOING TEST

**Fig 6.1 Real Time Test Monitoring**

**CHAPTER 5**

## **5.2 Database Design:**

The first step in the database design was to analyze the data that would be collected and determine the expected uses of the data. To aid in the management of the data and tables, each table would have at least one field that contained a unique identifier for that record, a field to identify who was doing the data entry, and another field to track when the data was entered. In most cases these fields are hidden from the user and are updated automatically by the system. This information is accessible by the administrator for troubleshooting purposes.

**Table 5.2.1: StudentDetails**

In this database table consists of the students details and login details.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** |
| StudentName | Varchar | 50 | Not Null |
| EmailId | Varchar | 50 | Primary key |
| Password | Varchar | 50 | Primary key |
| MobileNumber | Varchar | 50 | Not Null |
| BatchType | Varchar | 50 | Not Null |
| SId | Int | 20 | Not Null |
| SClass | Varchar | 20 | Not Null |
| Section | Varchar | 20 | Not Null |
| SchoolName | Varchar | 100 | Not Null |

**CHAPTER 5**

**Table 5.2.2: TestDetails:**

In this table, it consists details tests with having primary key as TestId and foreign key as SClass and BatchType

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** |
| TestId | Int | 20 | Primary key |
| TestName | Varchar | 50 | Not Null |
| DateofCreation | Datetime | Default | Not Null |
| NoofSubject | Int | 5 | Not Null |
| NoofQuestions | Int | 5 | Not Null |
| BatchType | Varchar | 50 | Foreign Key |
| Duration | Int | 10 | Not Null |
| ExamType | Varchar | 50 | Not Null |
| SClass | Varchar | 20 | Foreign Key |
| PositiveMark | Int | 10 | Not Null |
| NegativeMark | Int | 10 | Not Null |
| TotalMark | Int | 10 | Not Null |
| StartTime | Time | 7 | Not Null |
| IsCreatorVerify | Bit | Default | Not Null |
| IsTeacherVerify | Bit | Default | Not Null |
| IsAutoLaunch | Bit | Default | Not Null |
| IsTimeExceed | Bit | Default | Not Null |

**Table 5.2.3: Test Taken Status:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** |
| Sid | Int | 20 | Primary Key |
| TestId | Int | 20 | Primary Key |
| Status | Varchar | 50 | Not Null |
| LoginTime | datetime | Default | Not Null |

**Table 5.2.4: StudentPingTime**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** |
| TestId | Int | 20 | Primary Key |
| SId | Int | 20 | Primary Key |
| LastPingTime | Varchar | 100 | Not Null |

**CHAPTER 5**

**Table 5.2.5: CompletedReason**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** |
| TestId | Int | 20 | Primary Key |
| SId | Int | 20 | Primary Key |
| Reason | Varchar | 50 | Not Null |

## **Normalization**

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

Normalization is used for mainly two purposes,

* + - Eliminating redundant (useless) data.
    - Ensuring data dependencies make sense i.e. data is logically stored.

#### **Normalization Rule**

The following are the normal forms applied in this application,

* + - * First Normal Form
      * Second Normal Form

#### **First Normal Form (1NF)**

In the database as per the rule of first normal form (1NF), an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

#### **Second Normal Form (2NF)**

By introducing the primary key wherever it is necessary, the database tables will in second normal form (2NF).

**CHAPTER 5**

## **Third Normal Form (3NF)**

There are no transitive functional dependencies and hence our table is in 3NF.

## **DataBase Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Size** | **Description** |
| StudentName | Varchar | 50 | Name of the Student |
| EmailId | Varchar | 50 | Student Email Id |
| Password | Varchar | 50 | Password for Student to login |
| MobileNumber | Varchar | 50 | Mobile Number of Student |
| BatchType | Varchar | 50 | Student Batch Type |
| Sid | Int | 20 | Student Id |
| SClass | Varchar | 20 | Student Class |
| Section | Varchar | 20 | Student Class Section |
| SchoolName | Varchar | 20 | Name of the School |
| TestId | Int | 20 | Test Id |
| Status | Varchar | 50 | Student Status while Writing Test |
| LoginTime | Datetime | Default | Student Login Time |
| Reason | Varchar | 50 | Test Completed Reason |
| TestName | Varchar | 50 | Name of the Test |
| DateOfCreation | Datetime | Default | Test Creation Date |
| NoOfSubject | Int | 5 | Number of subject in test |
| NoOfQuestion | Int | 5 | Number of Question in test |
| Duration | Int | 5 | Test Duration |
| ExamType | Varchar | 50 | Type of Exam |
| PositiveMark | Int | 10 | Positive Mark for Question |
| NegativeMark | Int | 10 | Negative Mark for Question |
| TotalMark | Int | 10 | Total mark for Test |
| StartTime | Datetime | Default | Test Start Time |
| IsAutoLaunch | Bit | Default | Test is Auto Launch |
| IsTimeExceed | Bit | Default | Test date is exceeded |
| QNo | Int | 5 | Question Number |
| Question | Varchar | 100 | Question Details |
| Option | Varchar | 100 | Question Options |
| Answer | Varchar | 100 | Question Answer |

## **5.4. E-R Diagram:**

Password

Batch Type

SId

EmailID

SClass

StudentName

Section

StudentDetails

TestTakenStatus

SId

Status

TestId

SId

TestId

LoginTime

Reason

CompletedReason

TestDetails

TestName

NoofSubject

BatchType

Duration

TestId

NoofQuestions

DateofCreation

**CHAPTER 6**

# **SYSTEM DEVELOPMENT**

## **Special Feature of the Language:**

**WPF – Windows Presentation Foundation:**

**Windows Presentation Foundation** (**WPF**) is a free and open-source graphical subsystem (similar to WinForms) originally developed by Microsoft for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0 in 2006.

WPF employs XAML, an XML-based language, to define and link various interface elements. WPF applications can be deployed as standalone desktop programs or hosted as an embedded object in a website. WPF aims to unify a number of common user interface elements, such as 2D/3D rendering, fixed and adaptive documents, typography, vector graphics, runtime animation, and pre-rendered media. These elements can then be linked and manipulated based on various events, user interactions, and data bindings. WPF uses XAML as its frontend language and C# as its backend languages.

#### **XAML**

**XAML** stands for Extensible Application Markup Language. It is a declarative XML- based language developed by Microsoft that is used for initializing structured values and objects. It is mainly used for designing GUIs, however it can be used for other purposes as well, e.g., to declare workflow in Workflow Foundation.

XAML is used extensively in .NET Framework 3.0 & .NET Framework 4.0 technologies, particularly Windows Presentation Foundation (WPF), Silverlight, Windows Workflow Foundation (WF), Windows Runtime XAML Framework and Windows Store apps.

In WPF, XAML forms a user interface markup language to define UI elements, data binding, events, and other features. XAML can also be used in Silverlight applications, Windows 10 Mobile (previously Windows Phone) and Universal Windows Platform apps, also called Windows Store apps. Declare your UI in XAML has some advantages:

**CHAPTER 6**

* XAML code is short and clear to read
* Separation of designer code and logic
* Graphical design tools like Expression Blend require XAML as source.
* The separation of XAML and UI logic allows it to clearly separate the roles of designer and developer.

#### **C#(C SHARP)**

**C#** is a general-purpose, multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class- based), and component-oriented was developed around 2000 by Microsoft as part of its .NET initiative. C# was designed by Anders Hejlsberg. C# is used to develop web apps, desktop apps, mobile apps, games and much more.

Key characteristics of C# language includes,

* Modern and easy
* Fast and open source
* Cross platform
* Safe
* Versatile
* Evolving

#### **Microsoft SQL Server**

**Microsoft SQL Server** is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft SQL Server is one of the three market-leading database technologies, along with Oracle Database and IBM's DB2. Like other RDBMS software, Microsoft SQL Server is built on top of SQL, a standardized programming language that database administrators (DBAs) and other IT professionals use to manage databases and query the data they contain.

**CHAPTER 6**

SQL Server is tied to Transact-SQL (T-SQL), an implementation of SQL from Microsoft that adds a set of proprietary programming extensions to the standard language.

**SQL (Structured Query Language)**

* SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.
* SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Microsoft SQL Server, MYSQL etc.
* Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system.
* However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.
* SQL helps to generate useful strategies from a database and can easily interact with large and massive database, no matter what is the size.

These features make SQL a most powerful tool. Hence, here are some of the major SQL features which makes it a successful database programming language:

1. High Performance, Availability and Security
2. Scalability and Flexibility
3. Robust Transactional Support
4. Comprehensive Application Development
5. Management Ease
6. Open Source

**RDLC(REPORT DEFINITION LANGUAGE CLIENT):**

RDLC is Report Definition Language Client-side. It is the extension of report file. It is used to create report s using Microsoft reporting technology. These files are created by the Visual Studio 2005 version of Report Designer. RDLC reports can be executed directly by the Report Viewer control in client side.

1. RDLC files are created by the Visual Studio 2005 version of Report Designer
2. It is used in Visual Studio
3. It is local report
4. No need a Reporting Services instance

**CHAPTER 7**

# **IMPLEMENTATION AND TESTING**

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

**Types of Testing Done:**

* Unit Testing
* Integration Testing
* Validation Testing

**Unit Testing:**

All modules were tested and individually as soon as they were completed and were checked for their correct functionality. In all modules testing the process and loops and functions are processing correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Description** | **Action** | **Test Data** | **Expected Result** |
| 1 | To test whether each page is functioning properly | Opening page which disable for temporary | Commenting Certain XAML tags | Application will not move to that page |

**CHAPTER 7**

**Integration Testing:**

The entire project was split into small program; each of these single programs gives a frame as an output. These programs were tested individually at last all these programs where combined together by creating another program where all these constructors were used. It give a lot of problem by not functioning is an integrated manner. The user interface testing is important since the user has to declare that the arrangements made in frames are convenient and it is satisfied. When the frames where given for the test, the end user gave suggestion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Description** | **Action** | **Test Data** | **Expected Result** |
| 1 | To test whether each data is showing properly | Opening Each Module | Alter Database details | Application should display wrong data |

**Validation Testing:**

In each module the required field validation is tested. Validation succeeds when the software function in a manner that can be reasonably accepted by the user. Determining if the system complies with the requirements and performs functions for which it is intended and meets the organization’s goals and user needs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Id** | **Description** | **Action** | **Test Data** | **Expected Result** |
| 1 | To test whether error message is displayed or not when try to get report without selecting any batch | Opening completed module | Not Selecting Any batch | Application should display error “Please Select Any Batch” |

**CHAPTER 8**

# **CONCLUSION**

**Summary of the Project:**

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

* Automation of the entire system improves the efficiency
* It provides a friendly graphical user interface which proves to be better when compared to the existing system.
* It gives appropriate access to the authorized users depending on their permissions.
* It effectively overcomes the delay in communications.
* Updating of information becomes so easier.
* System security, data security and reliability are the striking features.
* The System has adequate scope for modification in future if it is necessary.

**Future Enhancements:**

* This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information through visualization regarding performance of the student in the school academics offered in **REAL TEST MONITORING**.
* The future enhancement is that can add the **ANOMALIES DETECTION COMPUTER BASED SYSTEM.** It will help to improve the result of each student. This is the most important future enhancement that could think of at present.
* Developing this application as an android application
* Adding more feature based on bugs, issues and customer requirements

# **BIBLIOGRAPHY**

#### **Book References**

* + - Abraham Silberschatz, “Database Systems”, McGraw Hill International, 1997.
    - Roger S Pressman, “Software Engineering”, McGraw Hill, International 6th Edition, New York, 2006.

#### **Web References**

1. **Windows Presentation Foundation (WPF)**

* <https://docs.microsoft.com/en-us/dotnet/framework/wpf/>

1. **C Sharp (C#):**

* <https://docs.microsoft.com/en-us/dotnet/csharp/>

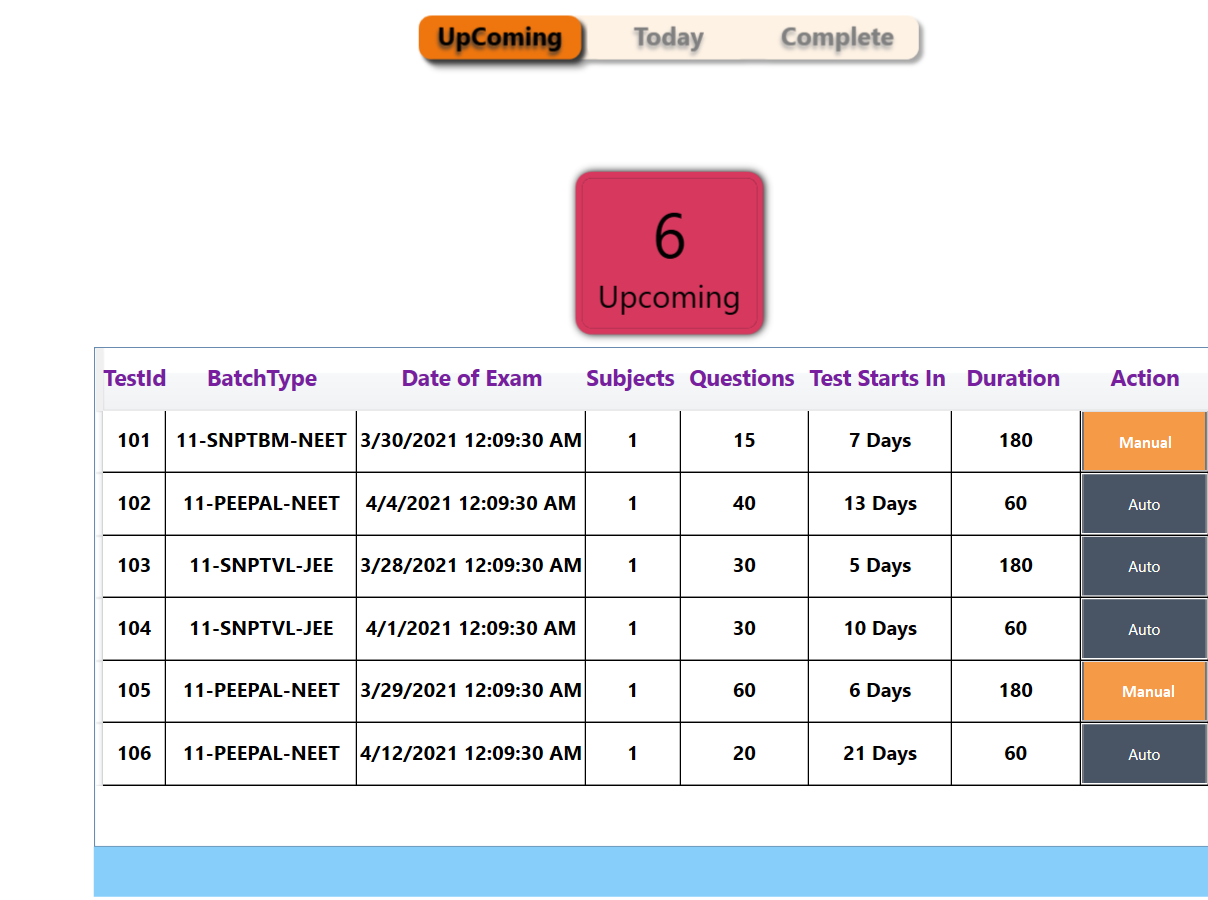
1. **MS SQL Server 2019**

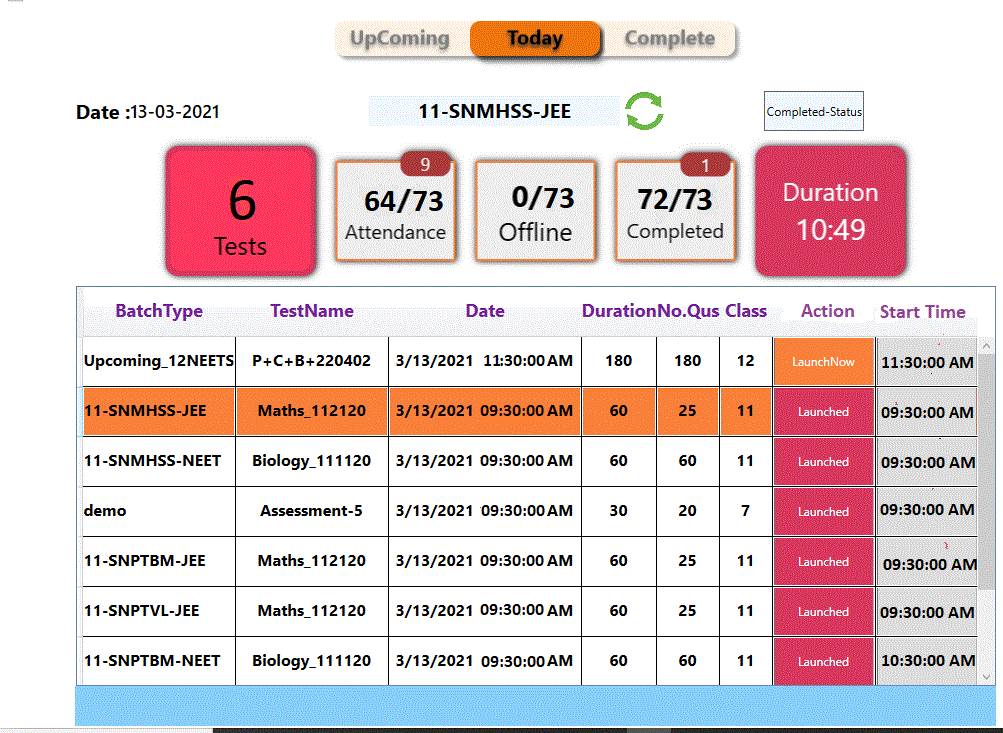
* <https://docs.microsoft.com/en-us/sql/2014-toc/?view=sql-server-2014>

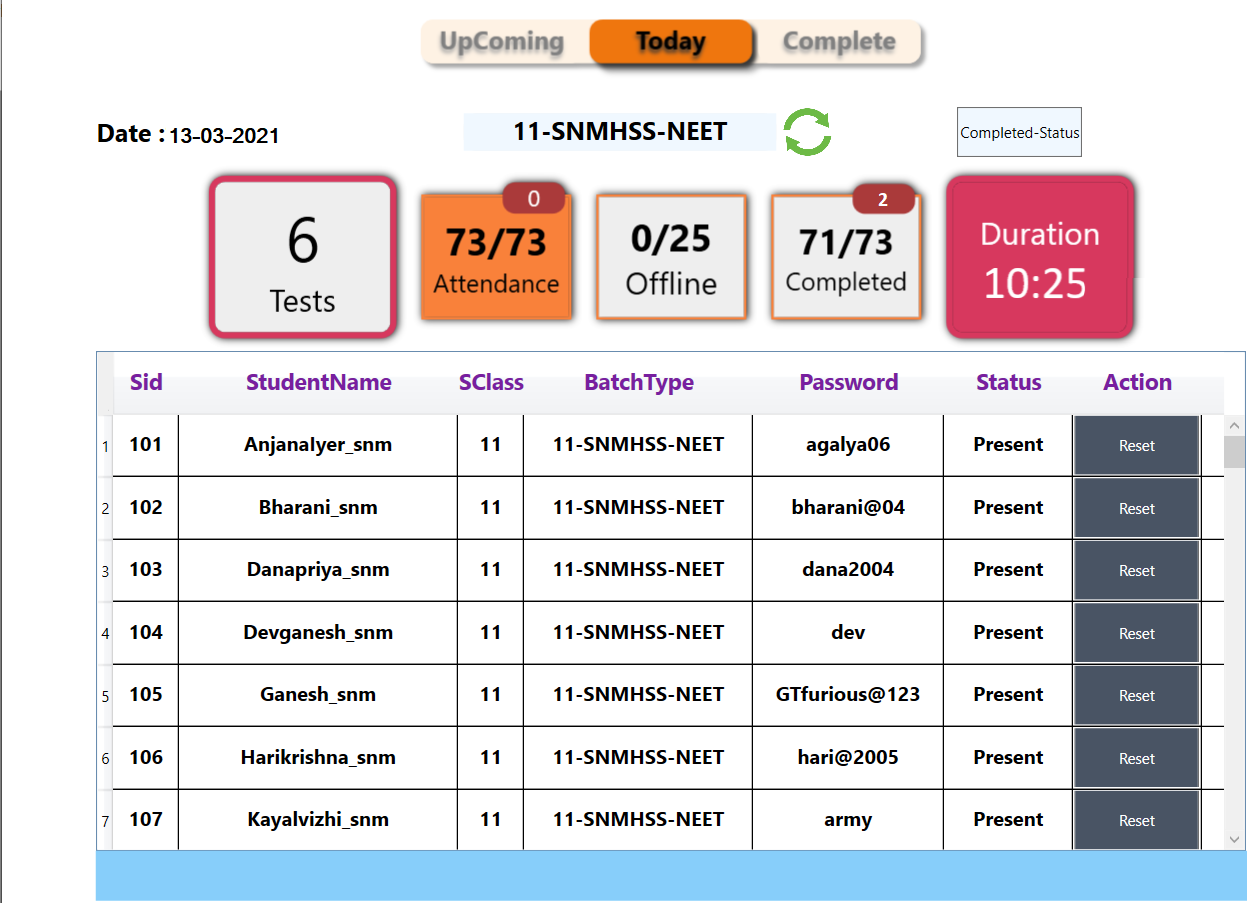
**ANNEXURE**

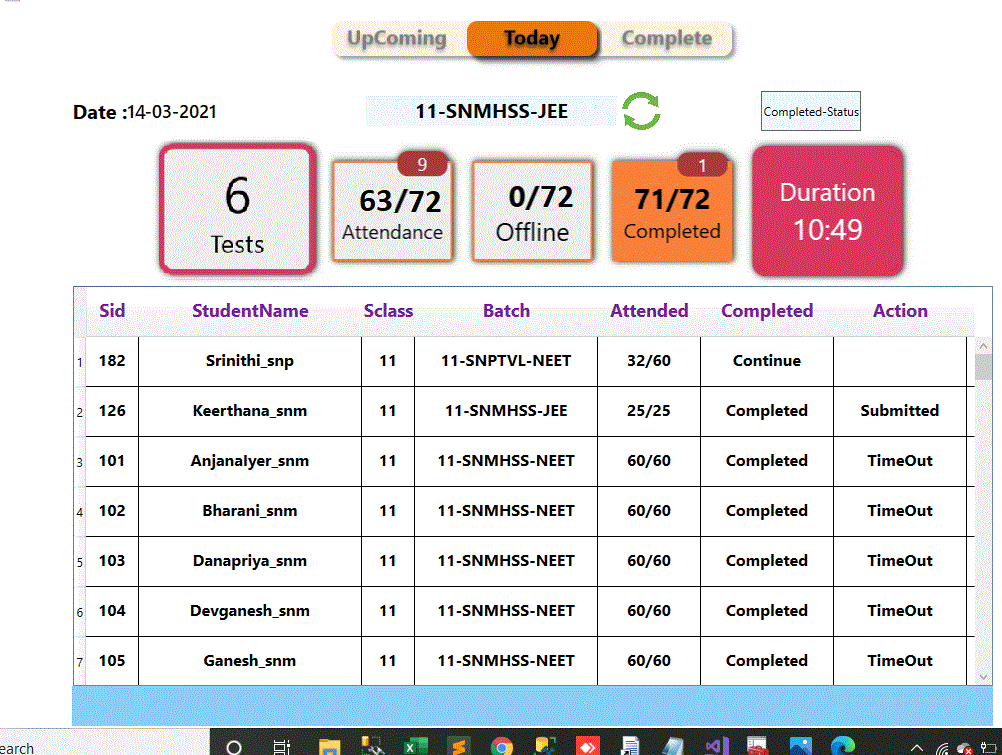
**USER INTERFACE SCREENSHOTS**

**Upcoming Dashboard:**

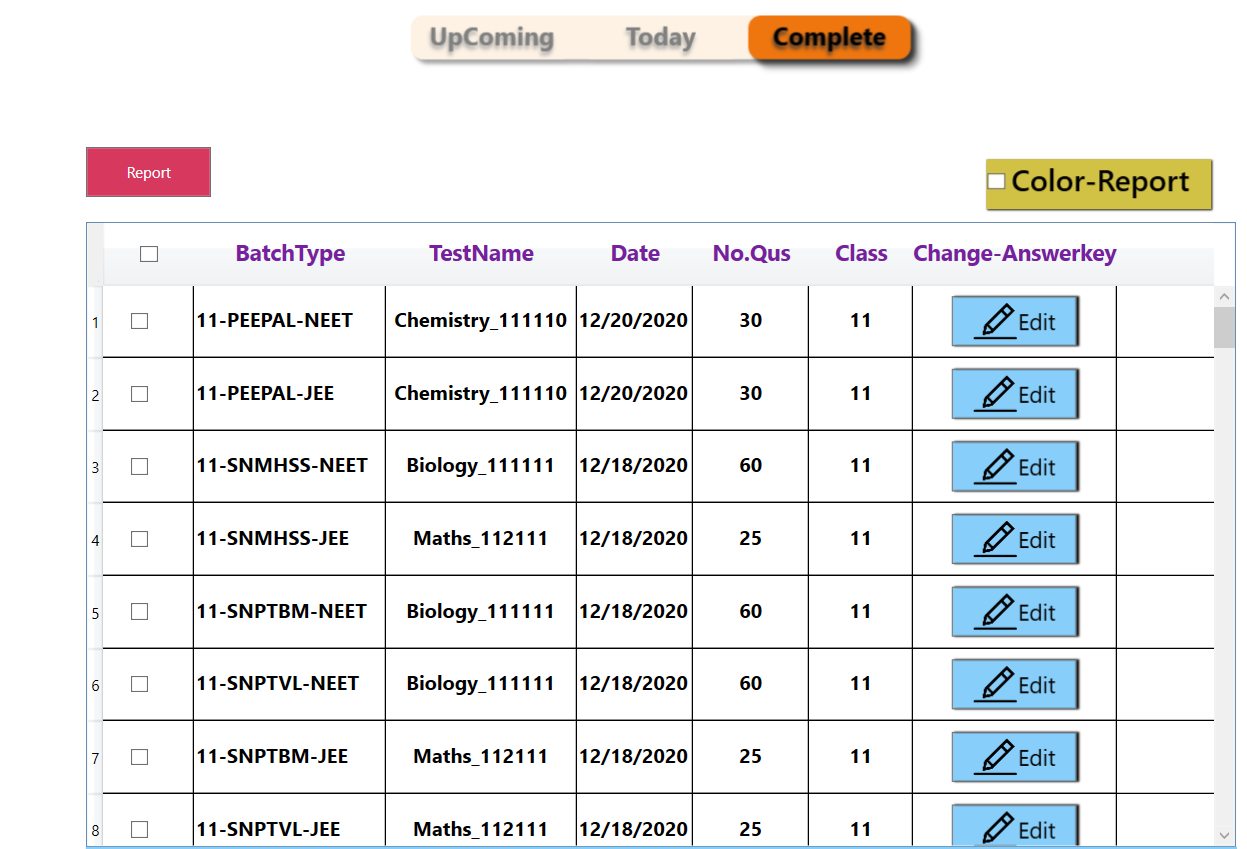
****

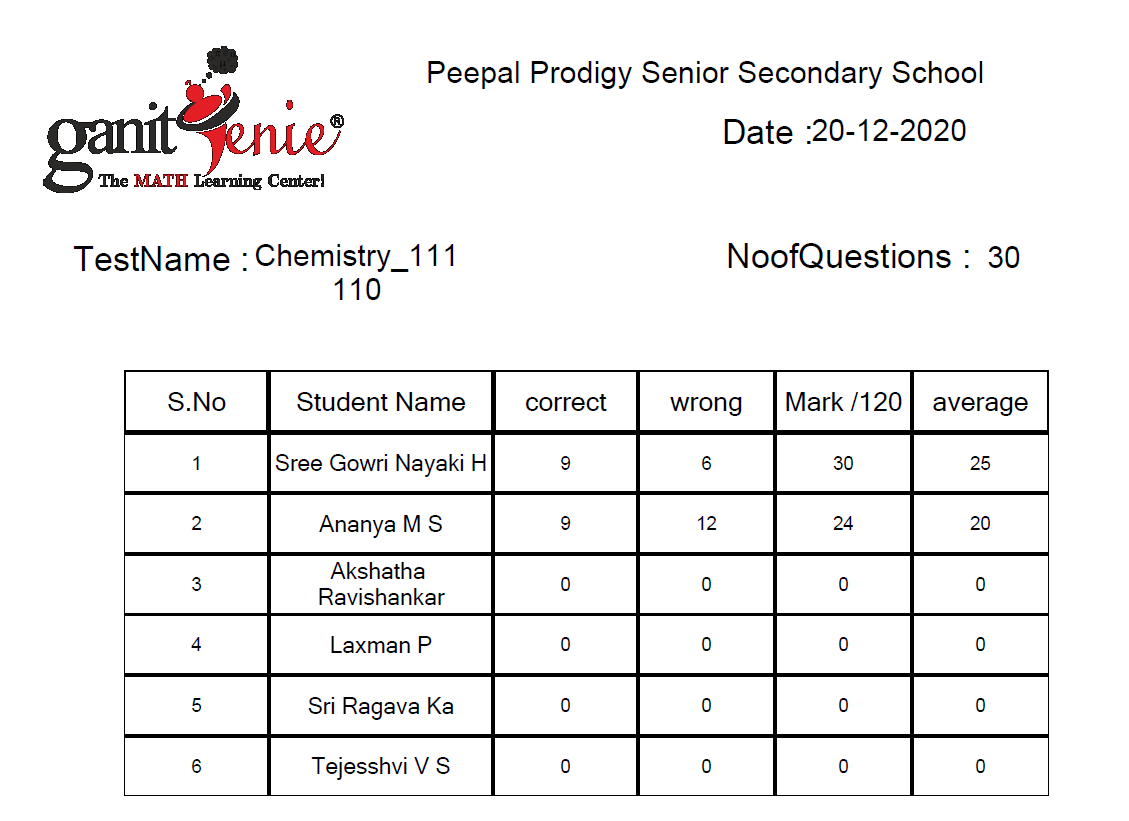
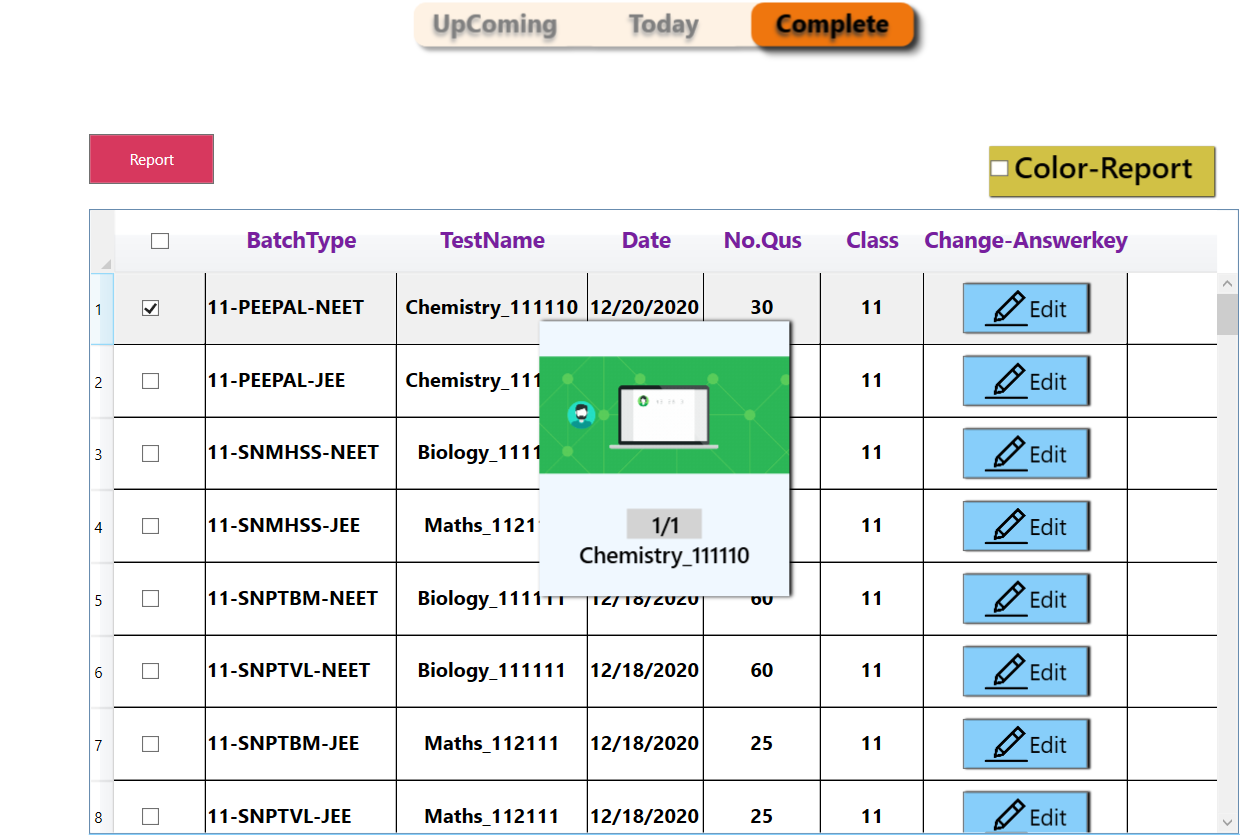
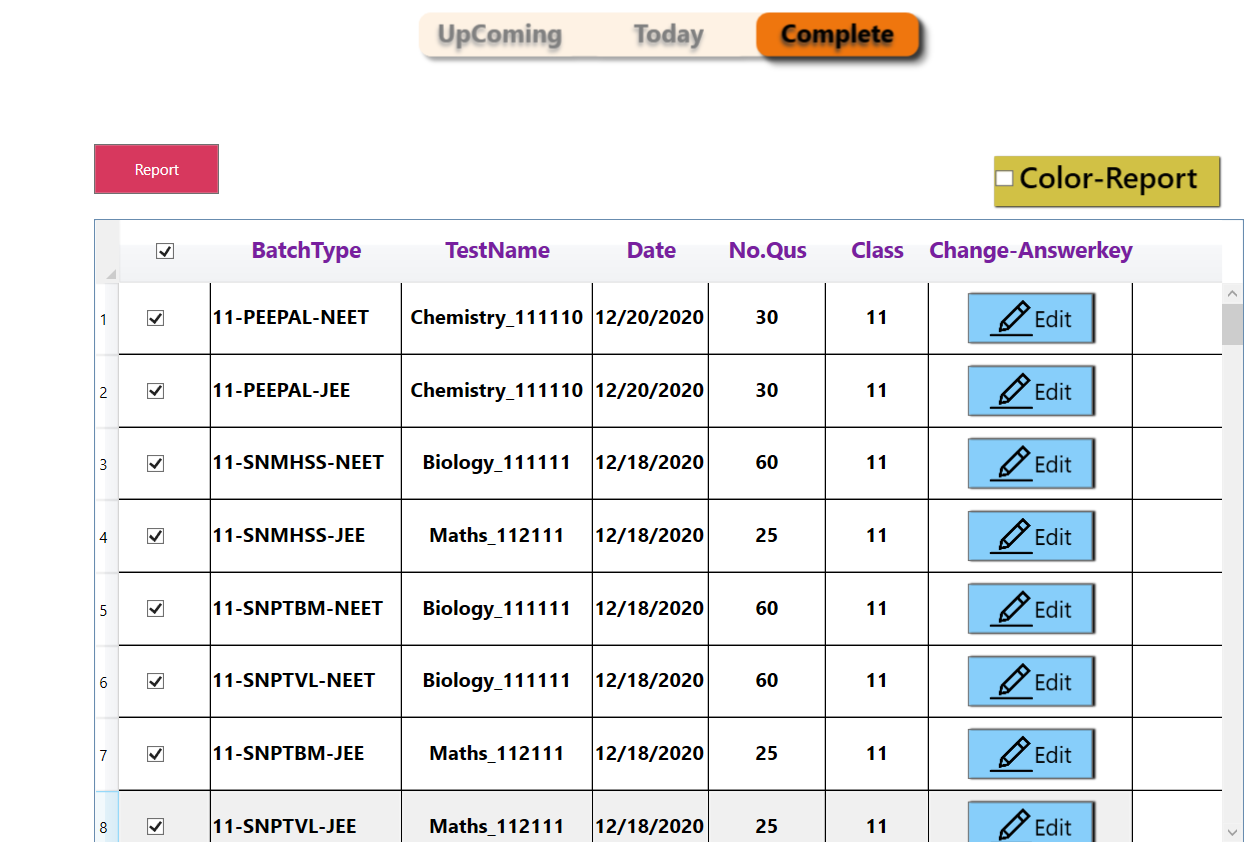
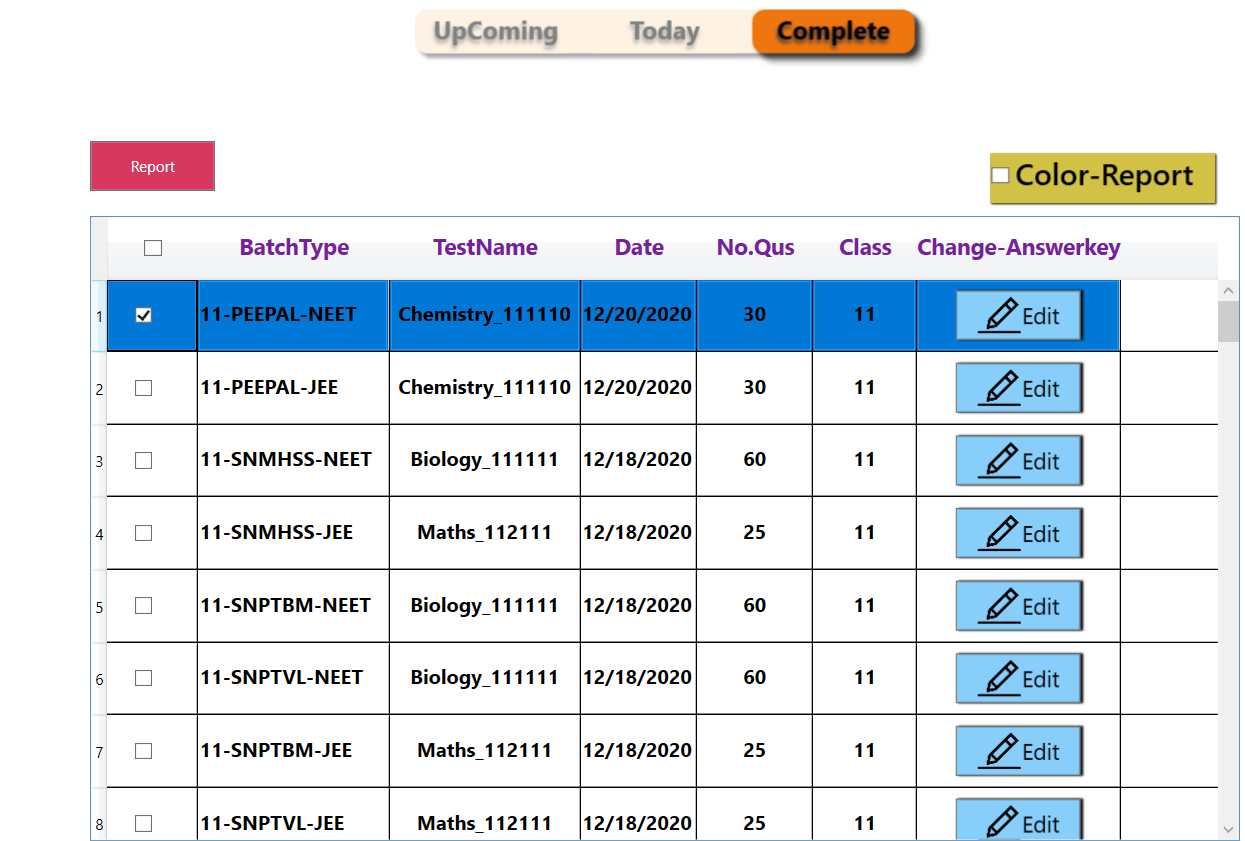
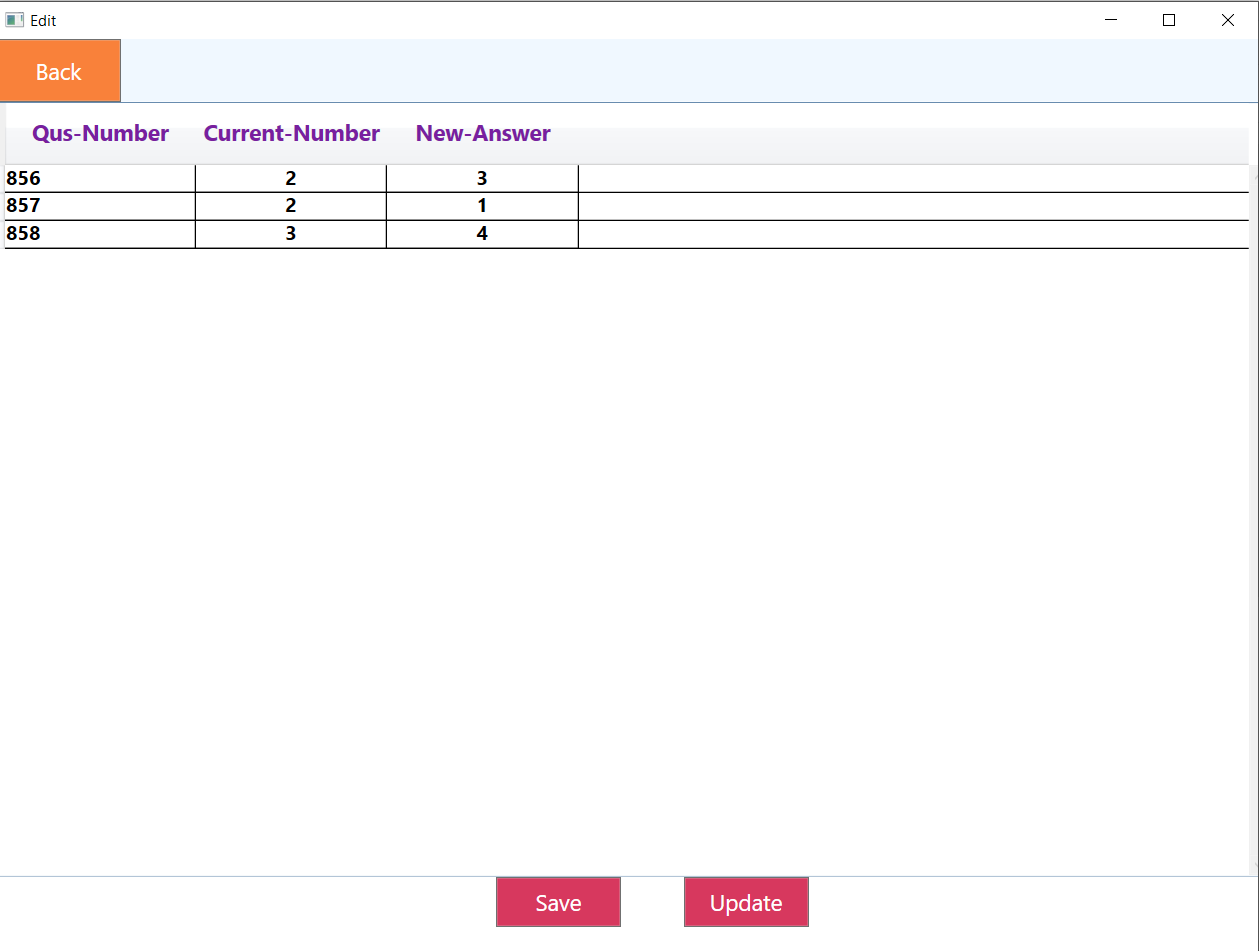
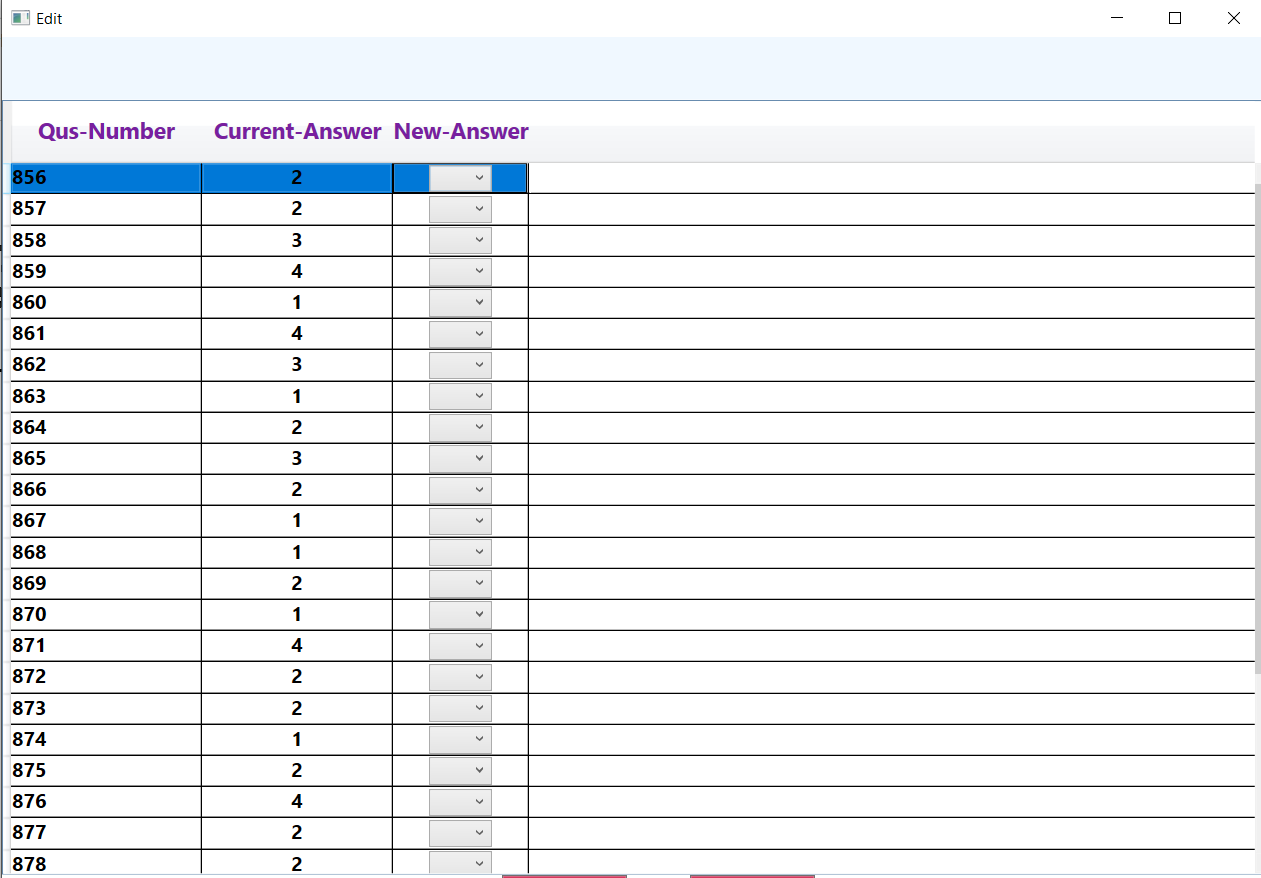
**Today Dashboard: **

****

****

**Completed Dashboard:**

****

****

**ANEXURE**

**SAMPLE CODES**

1. **SQL TABLE CREATION:**

create table **StudentDetails**(StudentName varchar(50) not null,

EmailId varchar(50) not null,

Password varchar(50) not null,

MobileNumber varchar(50) not null,

BatchType varchar(50) not null,

SId int identity(1,1),

SClass varhcar(20) not null,

Section varhcar(20) not null,

SchoolName varchar(100) not null,

primary key(EmailId,Password);

create table **TestDetails**(TestId int identity(1,1) not null,

TestName varchar(50) not null,

DateofCreation datetime not null,

NoofSubject int not null,

NoofQuestions int not null,

BatchType varchar(50) not null,

Duration int not null

,ExamType varchar(50) not null,

SClass varchar(50) not null,

PositiveMark int not null, NegativeMark int not null,

TotalMark int not null,

Start time not null,

IsCreatorVerify bit not null,

IsTeacherVerify bit not null,

IsAutoLaunch bit not null,

IsTimeExceed bit not null,

primary key(TestId)

create table **TestTakenStatus**(SId int not null,

TestId int not null, Status varchar(50) not null,

LoginTime varchar(100) not null,

primary key(SId,TestId),

foreign key(TestId) references TestDetails(TestId),

foreign key(SId) references StudentDetails(SId)

create table **StudentTestPingTime**(TestId int not null,

SId int not null,

LastPingTime varchar(100) not null,

primary key(TestId,SId),foreign key(TestId) references TestDetails(TestId),

foreign key(SId) references StudentDetails(SId)

create table **CompletedReason**(Testid int not null,

SId int not null,

Reason varchar(100) not null,

primary key(Testid,SId),

foreign key(Testid) references TestDetails(Testid),

foreign key(SId) references StudentDetails(SId)

1. **WPF:**

**Upcoming Page:**

<Page x:Class="OngoingPage1.App.View.Pages.UpComingTestDetails"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:gif="http://wpfanimatedgif.codeplex.com"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:OngoingPage1.App.View.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="920"

Title="UpComingTestDetails">

<Grid>

<Grid.Resources>

<Style TargetType="DataGridCell">

<Setter Property="Height" Value="50"></Setter>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightGray" />

</Trigger>

<Trigger Property="DataGridCell.IsSelected" Value="True">

<Setter Property="Background" Value="#f9813a" />

</Trigger>

</Style.Triggers>

</Style>

</Grid.Resources>

<Grid.RowDefinitions>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="150"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

</Grid.RowDefinitions>

<!--<StackPanel Grid.RowSpan="4" HorizontalAlignment="Center" Margin="0,250,0,0" VerticalAlignment="Center" Panel.ZIndex="2">

<Image x:Name="g2" gif:ImageBehavior.AnimatedSource="F:\Final1\OngoingPage1\OngoingPage1\OngoingPage1.App\View\Images\Ellipsis-1s-200px.gif" Height="150" Width="150" Visibility="Collapsed" ></Image>

--><!--<ProgressBar Height="50" IsIndeterminate="True" ></ProgressBar>--><!--

</StackPanel>

<StackPanel Grid.Row="0" Orientation="Horizontal">

<TextBlock Text="Date :" FontSize="20" FontWeight="Bold" VerticalAlignment="Center"></TextBlock>

<TextBlock x:Name="date1" Text="dd-mm-yyy" FontSize="18" FontWeight="SemiBold" VerticalAlignment="Center"></TextBlock>

<Border Height="30" Width="250" BorderBrush="#d7385e" Margin="150,0,0,0" Background="#eeeeee" CornerRadius="10">

<StackPanel Orientation="Vertical" Height="30" Width="250" Background="AliceBlue" >

<TextBlock x:Name="schoolname" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" Text="Root" FontSize="20" Width="250"></TextBlock>

</StackPanel>

</Border>

<StackPanel Height="50" x:Name="loading" Visibility="Visible">

--><!--<TextBlock Text="Loading" HorizontalAlignment="Center" FontSize="12" FontWeight="SemiBold"></TextBlock>--><!--

</StackPanel>

<Image Margin="100,0,0,0" x:Name="g1" gif:ImageBehavior.AnimatedSource="F:\Final1\OngoingPage1\OngoingPage1\OngoingPage1.App\View\Images\Ellipsis-1s-200px.gif" Height="30" Width="40" Visibility="Collapsed" ></Image>

<Image x:Name="Reload" Margin="5,0,0,0" Height="40" Source="F:\Final\Final Online Test Dash\OngoingPage1\OngoingPage1\OngoingPage1.App\View\Images\refresh.png" PreviewMouseLeftButtonDown="Image\_PreviewMouseLeftButtonDown">

</Image>

</StackPanel>-->

<StackPanel Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" Orientation="Horizontal" HorizontalAlignment="Center">

<Border x:Name="Test" Height="130" Width="150" BorderBrush="#d7385e" CornerRadius="10" Background="#d7385e" VerticalAlignment="Center" Margin="10" MouseDown="upcomingtests" >

<Border.Style>

<Style TargetType="Border">

<Setter Property="BorderThickness" Value="5"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderThickness" Value="5"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<Border.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="1" Direction="1.0" Opacity="1.0" ></DropShadowEffect>

</Border.Effect>

<StackPanel VerticalAlignment="Center">

<TextBlock Text="6" x:Name="Testcount" FontSize="50" VerticalAlignment="Center" HorizontalAlignment="Center"></TextBlock>

<TextBlock Text="Upcoming" FontSize="25" VerticalAlignment="Bottom" HorizontalAlignment="Center"></TextBlock>

</StackPanel>

</Border>

</StackPanel>

<DataGrid Grid.Row="2" Height="400" Width="920" IsReadOnly="True" x:Name="scheduledgrid" AutoGenerateColumns="False" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

ScrollViewer.VerticalScrollBarVisibility="Visible" Background="#FFFFFF" >

<DataGrid.Columns>

<DataGridTemplateColumn Header="TestId" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="30" Text="{Binding TestId}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="BatchType">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="150" Text="{Binding BatchType}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Date of Exam">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="180" Text="{Binding DateofCreation}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Subjects " >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="20" Text="{Binding NoofSubject}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Questions ">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="100" Text="{Binding NoofQuestions}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black" ToolTipService.ShowDuration="5000000">

<TextBlock.ToolTip>

<StackPanel>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"></ColumnDefinition>

<ColumnDefinition Width="100"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="TestName"></TextBlock>

</Grid>

</StackPanel>

</TextBlock.ToolTip>

</TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Test Starts In " >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="100" Text="{Binding TestStartsIn}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Duration " >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="100" Text="{Binding Duration}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Action">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Height="50" Width="100" VerticalAlignment="Center" HorizontalAlignment="Center" Background="#495464" Content="Auto" Foreground="White"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

**Today Page:**

<Page x:Class="OngoingPage1.App.View.Pages.OngoinPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:gif="http://wpfanimatedgif.codeplex.com"

xmlns:local="clr-namespace:OngoingPage1.App.View.Pages"

mc:Ignorable="d"

d:DesignHeight="600" d:DesignWidth="920"

Title="OngoinPage">

<!--<Page.Resources>

--><!--Start--><!--

<ControlTemplate TargetType="{x:Type Button}" x:Key="button1">

<Border Background="#a3b1c7" Width="90" Height="25" CornerRadius="4" ToolTip="Click to Launch test now">

<Border.Effect>

<DropShadowEffect ShadowDepth="0.5"></DropShadowEffect>

</Border.Effect>

<Grid >

<TextBlock Text="Launch Now" Foreground="DarkSlateBlue" FontWeight="SemiBold" HorizontalAlignment="Center" VerticalAlignment="Center"></TextBlock>

</Grid>

</Border>

</ControlTemplate>

<Style TargetType="{x:Type Button}" x:Key="Sbutton1">

<Setter Property="Template" Value="{StaticResource button1}"/>

</Style>

--><!--End-->

<!--Start--><!--

<ControlTemplate TargetType="{x:Type Button}" x:Key="button2">

<Border Background="#a3b1c7" Width="90" Height="25" CornerRadius="4" ToolTip="Click to Launch test now">

<Border.Effect>

<DropShadowEffect ShadowDepth="0.5"></DropShadowEffect>

</Border.Effect>

<Grid>

<TextBlock Text="Launch Now" Foreground="Red" FontWeight="SemiBold" HorizontalAlignment="Center" VerticalAlignment="Center"></TextBlock>

</Grid>

</Border>

</ControlTemplate>

<Style TargetType="{x:Type Button}" x:Key="Sbutton2">

<Setter Property="Template" Value="{StaticResource button2}"/>

</Style>

--><!--End--><!--

</Page.Resources>-->

<Grid>

<Grid.Resources>

<Style TargetType="DataGridCell">

<Setter Property="Height" Value="50"></Setter>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="LightGray" />

</Trigger>

<Trigger Property="DataGridCell.IsSelected" Value="True">

<Setter Property="Background" Value="#f9813a" />

</Trigger>

</Style.Triggers>

</Style>

</Grid.Resources>

<Grid.RowDefinitions>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="150"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

</Grid.RowDefinitions>

<Border x:Name="Warning" Height="120" Width="400" Grid.Row="1" Panel.ZIndex="2" BorderBrush="Red" CornerRadius="4" BorderThickness="3" Visibility="Collapsed" >

<StackPanel Background="White" >

<StackPanel Background="White">

<StackPanel.Effect>

<DropShadowEffect BlurRadius="3" ShadowDepth="2" Direction="1.0" Opacity="2"></DropShadowEffect>

</StackPanel.Effect>

<TextBlock Text="Do You Want to Continue " FontSize="24" FontWeight="SemiBold" Foreground="Black" HorizontalAlignment="Center"></TextBlock>

</StackPanel>

<StackPanel Orientation="Horizontal" Background="LightBlue">

<StackPanel.Effect>

<DropShadowEffect BlurRadius="3" ShadowDepth="2" Direction="1.0" Opacity="2"></DropShadowEffect>

</StackPanel.Effect>

<Button Height="50" VerticalAlignment="Center" Content="Cancel" FontFamily="roboto" FontSize="18" Foreground="White" Margin="20" Background="#d7385e" Width="120" Click="Cancelclick"></Button>

<Button Height="50" VerticalAlignment="Center" Content="Confirm" FontFamily="roboto" FontSize="18" Margin="80,0,0,0" Foreground="White" Background="#0e49b5" Width="120" Click="AllCompletedClick"></Button>

</StackPanel>

</StackPanel>

</Border>

<StackPanel Grid.RowSpan="4" HorizontalAlignment="Center" Margin="0,250,0,0" VerticalAlignment="Center" Panel.ZIndex="2">

<Image x:Name="g2" gif:ImageBehavior.AnimatedSource="/View/Images/Ellipsis-1s-200px.gif" Height="150" Width="150" Visibility="Collapsed" ></Image>

<StackPanel x:Name="l2" Visibility="Collapsed" Margin="0,-60,0,0">

<Image gif:ImageBehavior.AnimatedSource="/View/Images/launch.png" Height="150" Width="150" ></Image>

<TextBlock Text="Launching" FontSize="25" HorizontalAlignment="Center" FontFamily="ROBOTO" FontWeight="SemiBold" Foreground="#d7385e"></TextBlock>

</StackPanel>

<!--<ProgressBar Height="50" IsIndeterminate="True" ></ProgressBar>-->

</StackPanel>

<StackPanel Grid.Row="0" Orientation="Horizontal">

<TextBlock Text="Date :" FontSize="20" FontWeight="Bold" VerticalAlignment="Center"></TextBlock>

<TextBlock x:Name="date1" Text="dd-mm-yyy" FontSize="18" FontWeight="SemiBold" VerticalAlignment="Center"></TextBlock>

<Border Height="30" Width="250" BorderBrush="#d7385e" Margin="150,0,0,0" Background="#eeeeee" CornerRadius="10">

<StackPanel Orientation="Vertical" Height="30" Width="250" Background="AliceBlue" >

<TextBlock x:Name="schoolname" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" Text="Root" FontSize="20" Width="250"></TextBlock>

</StackPanel>

</Border>

<Image Margin="100,0,0,0" x:Name="g1" gif:ImageBehavior.AnimatedSource="/View/Images/Ellipsis-1s-200px.gif" Height="30" Width="40" Visibility="Collapsed" ></Image>

<Image x:Name="Reload" Margin="5,0,0,0" Height="40" Source="/View/Images/refresh.png" PreviewMouseLeftButtonDown="Image\_PreviewMouseLeftButtonDown">

</Image>

<Button x:Name="Completed" Margin="100,0,0,0" Content="Completed-Status" Background="AliceBlue" Height="40" Width="100" Click="Warningclick"></Button>

</StackPanel>

<StackPanel Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" Orientation="Horizontal" HorizontalAlignment="Center">

<Border x:Name="Test" Height="130" Width="150" BorderBrush="#d7385e" CornerRadius="10" Background="#d7385e" VerticalAlignment="Center" Margin="10" MouseDown="Testdetails" >

<Border.Style>

<Style TargetType="Border">

<Setter Property="BorderThickness" Value="5"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderThickness" Value="5"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<Border.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="1" Direction="1.0" Opacity="1.0" ></DropShadowEffect>

</Border.Effect>

<StackPanel VerticalAlignment="Center">

<TextBlock Text="0/0" x:Name="Testcount" FontSize="50" VerticalAlignment="Center" HorizontalAlignment="Center"></TextBlock>

<TextBlock Text="Tests" FontSize="25" VerticalAlignment="Bottom" HorizontalAlignment="Center"></TextBlock>

</StackPanel>

</Border>

<Button x:Name="Border1" Height="100" Width="120" BorderBrush="#f9813a" Background="#eeeeee" VerticalAlignment="Center" Margin="10" Click="Logindata" >

<Button.Style>

<Style TargetType="Button">

<Setter Property="BorderThickness" Value="2"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderThickness" Value="5"/>

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="1" Direction="1.0" Opacity="1.0" ></DropShadowEffect>

</Button.Effect>

<StackPanel VerticalAlignment="Center" >

<StackPanel Height="25" Width="50" Margin="60,-28,0,0">

<Border x:Name="absentvisible" Height="25" Width="50" Background="#aa3a3a" CornerRadius="10" Visibility="Collapsed">

<TextBlock x:Name="AbsentCount" Text="0" Foreground="White" FontSize="18" HorizontalAlignment="Center">

</TextBlock>

<Border.ToolTip>

<TextBlock Text="AbsentCount" FontSize="14" FontWeight="SemiBold"></TextBlock>

</Border.ToolTip>

</Border>

</StackPanel>

<TextBlock x:Name="TestLogin" Text="0/0" FontSize="30" FontWeight="Bold" HorizontalAlignment="Center" >

</TextBlock>

<TextBlock Text="Attendance" FontSize="20" VerticalAlignment="Bottom" HorizontalAlignment="Center"></TextBlock>

</StackPanel>

</Button>

<Button x:Name="Ongo" Height="100" Width="120" BorderBrush="#f9813a" Background="#eeeeee" VerticalAlignment="Center" Margin="10" Click="Ongo\_MouseDown" >

<Button.Style>

<Style TargetType="Button">

<Setter Property="BorderThickness" Value="2"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderThickness" Value="5"/>

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="1" Direction="1.0" Opacity="1.0" ></DropShadowEffect>

</Button.Effect>

<StackPanel x:Name="Ongoingpanel" VerticalAlignment="Center" >

<StackPanel Height="25" Width="50" Margin="60,-28,0,0">

<Border x:Name="pingborder" Background="#aa3a3a" CornerRadius="10" Visibility="Visible">

<TextBlock Name="pingtimebox" Text="0" Foreground="White" FontSize="18" HorizontalAlignment="Center">

</TextBlock>

<Border.ToolTip>

<TextBlock Text="OfflineCount" FontSize="14" FontWeight="SemiBold"></TextBlock>

</Border.ToolTip>

</Border>

</StackPanel>

<TextBlock x:Name="offlinecount" Text="0/0" FontSize="30" FontWeight="Bold" HorizontalAlignment="Center" ></TextBlock>

<TextBlock Text="Offline" FontSize="25" VerticalAlignment="Bottom" HorizontalAlignment="Center"></TextBlock>

</StackPanel>

</Button>

<Button x:Name="complete" Height="100" Width="120" BorderBrush="#f9813a" Background="#eeeeee" VerticalAlignment="Center" Margin="10" Click="CompletedClick" >

<Button.Style>

<Style TargetType="Button">

<Setter Property="BorderThickness" Value="2"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="BorderThickness" Value="5"/>

</Trigger>

</Style.Triggers>

</Style>

</Button.Style>

<Button.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="1" Direction="1.0" Opacity="1.0" ></DropShadowEffect>

</Button.Effect>

<StackPanel ToolTipService.ShowDuration="500000" VerticalAlignment="Center" >

<StackPanel Height="25" Width="50" Margin="60,-28,0,0">

<Border x:Name="tabborder" Background="#aa3a3a" Visibility="Collapsed" CornerRadius="10">

<TextBlock x:Name="tabcount" Text="0" Foreground="White" FontSize="18" HorizontalAlignment="Center"></TextBlock>

<Border.ToolTip>

<TextBlock Text="Ongoing Student count" FontSize="14" FontWeight="SemiBold"></TextBlock>

</Border.ToolTip>

</Border>

</StackPanel>

<StackPanel VerticalAlignment="Center">

<TextBlock x:Name="Copmletedstudents" Text="0/0" FontSize="28" FontWeight="Bold" HorizontalAlignment="Center" ></TextBlock>

<TextBlock Text="Completed" FontSize="20" VerticalAlignment="Bottom" HorizontalAlignment="Center"></TextBlock>

</StackPanel>

<!--<StackPanel.ToolTip >

<StackPanel>

<DataGrid Grid.Row="2" Width="200" IsReadOnly="True" x:Name="mysingledaygrid1" AutoGenerateColumns="False" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

ScrollViewer.VerticalScrollBarVisibility="Visible" Background="#FFFFFF" >

<DataGrid.Columns>

<DataGridTemplateColumn Header="List Of Students" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#f9813a"></Setter>

<Setter Property="Height" Value="30"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="15"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Height="20" Width="200" Text="{Binding .}" TextAlignment="Center" FontWeight="Bold" FontSize="15" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</StackPanel>

</StackPanel.ToolTip>-->

</StackPanel>

</Button>

<Border Height="130" Width="150" BorderBrush="#d7385e" Background="#d7385e" CornerRadius="10" VerticalAlignment="Center" Margin="10" >

<Border.Style>

<Style TargetType="Border">

<Setter Property="BorderThickness" Value="5"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#d7385e"/>

</Trigger>

</Style.Triggers>

</Style>

</Border.Style>

<Border.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="1" Direction="1.0" Opacity="1.0" ></DropShadowEffect>

</Border.Effect>

<StackPanel VerticalAlignment="Center">

<TextBlock Text="Duration" FontSize="25" Foreground="White" VerticalAlignment="Bottom" HorizontalAlignment="Center"></TextBlock>

<TextBlock x:Name="timeclose" Text="00:00:00" FontSize="30" Foreground="White" HorizontalAlignment="Center" ></TextBlock>

</StackPanel>

</Border>

</StackPanel>

<DataGrid Grid.Row="2" Height="400" Width="920" IsReadOnly="True" x:Name="Studentgrid" AutoGenerateColumns="False" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

ScrollViewer.VerticalScrollBarVisibility="Visible" Background="#FFFFFF" LoadingRow="Studentgrid\_LoadingRow" >

<DataGrid.Columns>

<DataGridTemplateColumn Header="Sid" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="50" Text="{Binding SId}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="StudentName">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="220" Text="{Binding EmailId}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="SClass">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="50" Text="{Binding Sclass}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="BatchType" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="180" Text="{Binding BatchType}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Password">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="150" Text="{Binding Password}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black" ToolTipService.ShowDuration="5000000">

<!--<TextBlock.ToolTip>

<StackPanel>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"></ColumnDefinition>

<ColumnDefinition Width="100"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="TestName"></TextBlock>

</Grid>

</StackPanel>

</TextBlock.ToolTip>-->

</TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Status" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="100" Text="{Binding Status}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Action">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Height="50" Width="100" VerticalAlignment="Center" HorizontalAlignment="Center" Background="#495464" Content="Reset" Foreground="White" Click="Button\_Click" ></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<!--//grid load-->

<DataGrid Grid.Row="2" Visibility="Collapsed" Width="920" Height="400" IsReadOnly="True" x:Name="OnGoingGrid" AutoGenerateColumns="False" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

ScrollViewer.VerticalScrollBarVisibility="Visible" Background="#FFFFFF" >

<DataGrid.Columns>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="80" Text="{Binding sid}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="200" Text="{Binding Name}" TextWrapping="Wrap" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black" ToolTipService.ShowDuration="5000000">

<TextBlock.ToolTip>

<StackPanel>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"></ColumnDefinition>

<ColumnDefinition Width="100"></ColumnDefinition>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

<RowDefinition Height="50"></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="TestName"></TextBlock>

</Grid>

</StackPanel>

</TextBlock.ToolTip>

</TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="80" Text="{Binding sclass}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="80" Text="{Binding Testid}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="160" Text="{Binding mobileno}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="160" Text="{Binding lasttime}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Height="40" Width="130" Text="{Binding schlname}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<DataGrid Grid.Row="2" Height="400" Width="920" IsReadOnly="True" x:Name="Studentgrid1" AutoGenerateColumns="False" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

ScrollViewer.VerticalScrollBarVisibility="Visible" Background="#FFFFFF" LoadingRow="Studentgrid\_LoadingRow" >

<DataGrid.Columns>

<DataGridTemplateColumn Header="Sid"

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate

<DataTemplate>

<TextBlock Width="50" Text="{Binding SId}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black" ToolTipService.ShowDuration="5000000">

</TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="StudentName">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="220" Text="{Binding StudentName}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Sclass">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="50" Text="{Binding Sclass}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black" ToolTipService.ShowDuration="5000000">

</TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Batch" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="180" Text="{Binding BatchType}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Attended" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="100" Text="{Binding AttendQus}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Completed" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="130" Text="{Binding TabClose}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Action">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="130" Text="{Binding reason}" VerticalAlignment="Center" HorizontalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<DataGrid Grid.Row="2" Height="400" Width="920" IsReadOnly="True" x:Name="Testdatagrid" AutoGenerateColumns="False" ScrollViewer.HorizontalScrollBarVisibility="Disabled"

ScrollViewer.VerticalScrollBarVisibility="Visible" Background="#FFFFFF" SelectedCellsChanged="Testdatagrid\_SelectedCellsChanged" VerticalAlignment="Bottom" >

<DataGrid.Columns>

<DataGridTemplateColumn Header="BatchType" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock VerticalAlignment="Center" Width="150" Text="{Binding BatchType}" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="TestName" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Text="{Binding TestName}" Width="150" VerticalAlignment="Center" TextAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<!--<DataGridTemplateColumn Header="TestId">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="30" Text="{Binding TestId}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black" ToolTipService.ShowDuration="5000000">

</TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>-->

<DataGridTemplateColumn Header="Date">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="190" Text="{Binding DateofCreation}" TextAlignment="Center" VerticalAlignment="Center" HorizontalAlignment="Center" FontWeight="Bold" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Duration">

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="60" Text="{Binding Duration}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="No.Qus" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="60" Text="{Binding NoofQuestions}" TextAlignment="Center" FontWeight="Bold" VerticalAlignment="Center" HorizontalAlignment="Center" FontSize="16" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Class" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="50" Text="{Binding SClass}" TextAlignment="Center" FontWeight="Bold" FontSize="16" VerticalAlignment="Center" HorizontalAlignment="Center" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="StartTime" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<TextBlock Width="100" Text="{Binding TimerString}" TextAlignment="Center" FontWeight="Bold" FontSize="16" VerticalAlignment="Center" HorizontalAlignment="Center" FontFamily="Roboto" Foreground="Black"></TextBlock>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn Header="Autolaunch" >

<DataGridTemplateColumn.HeaderStyle>

<Style TargetType="{x:Type DataGridColumnHeader}">

<Setter Property="Foreground" Value="#781F9F"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="HorizontalAlignment" Value="Center"></Setter>

<Setter Property="FontSize" Value="18"></Setter>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="FontFamily" Value="Roboto"></Setter>

<Setter Property="FontWeight" Value="Bold" />

<Setter Property="Background" Value="Transparent"></Setter>

</Style>

</DataGridTemplateColumn.HeaderStyle>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Height="50" Width="100" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="White" Click="LaunchNow" >

<Button.Style>

<Style TargetType="Button">

<Style.Triggers>

<DataTrigger Binding="{Binding launched}" Value="True">

<Setter Property="Content" Value="LaunchNow"></Setter>

<Setter Property="Background" Value="#f9813a"></Setter>

</DataTrigger>

<DataTrigger Binding="{Binding launched}" Value="False">

<Setter Property="Content" Value="Launched"></Setter>

<Setter Property="Background" Value="#d7385e"></Setter>

<Setter Property="Foreground" Value="Black"></Setter>

</DataTrigger>

</Style.Triggers>

</Style>

</Button.Style>

</Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

</Grid>

</Page>

**C#(C Sharp):**

**TodayPage:**

public partial class OngoinPage : Page

{

IStudentDetailRepository istud = new StudentdetailsRepository();

IcompleteDetailsRepository itest = new CompletedReasonDetails();

Itestdetailsrepository icurrentdaytest = new TestDetailsRepoistory();

DispatcherTimer Time = new DispatcherTimer();

List<TestDetails> te = new List<TestDetails>();

List<SubmitedReason> Tab = new List<SubmitedReason>();

List<SubmitedReason> LOG = new List<SubmitedReason>();

SortedList<string, List<SubmitedReason>> Tab1 = new SortedList<string, List<SubmitedReason>>();

SortedList<string, List<SubmitedReason>> LOG1 = new SortedList<string, List<SubmitedReason>>();

SortedList<string, string> closetiming = new SortedList<string, string>();

List<TestDetails> autolaunching = new List<TestDetails>();

List<SubmitedReason> Overalldata = new List<SubmitedReason>();

ObservableCollection<TestDetails> currentGridDataTimer = new ObservableCollection<TestDetails>();

SortedList<Int32, TestDetails> currenttest = new SortedList<Int32, TestDetails>();

int Actual;

int Attend;

int Present;

int time1;

int timer2 = 0;

bool Tagging = false;

int ActualStudents;

int localtemp;

int localtemp1;

int localtemp2;

string batch;

int testid;

//Ping Time List

#region

SortedList<int, List<TestPingTime>> pingtimevisible = new SortedList<int, List<TestPingTime>>();

List<TestPingTime> countlist = new List<TestPingTime>();

List<griddata> startlogintime = new List<griddata>();

SortedList<int, List<TestPingTime>> gettime = new SortedList<int, List<TestPingTime>>();

public void initializeload()

{

currenttest = icurrentdaytest.TestDetailsRepoistory(GetDateString(Convert.ToDateTime(today)));

te = currenttest.Values.ToList();

foreach (var j in te)

{

ActualStudents = ActualStudents + istud.GetStudentsinBatchtype(j.BatchType).Count();

currentGridDataTimer.Add(j);

if (j.IsAutoLaunch == true)

{

autolaunching.Add(new TestDetails { BatchType = j.BatchType, TestId = j.TestId, StartTime = j.StartTime, IsAutoLaunch = j.IsAutoLaunch });

}

}

preloaddata();

}

public void preloaddata()

{

//loading.Visibility = Visibility.Visible;

Tab1.Clear();

LOG1.Clear();

countlist.Clear();

pingtimevisible.Clear();

closetiming.Clear();

Overalldata.Clear();

gettime.Clear();

foreach (var s in currenttest)

{

List<SubmitedReason> temp = new List<SubmitedReason>();

List<SubmitedReason> temp1 = new List<SubmitedReason>();

temp1 = itest.GetTestAbsentStudents(s.Value.BatchType, s.Key);

temp = itest.GetTestTabclosedStudets(s.Value.BatchType, s.Key);

string temp3 = itest.gettestclosingtime(s.Key, s.Value.Duration);

temp1.Where(x => x.SId != ' ').Select(x => { x.testid = s.Key; return x; }).ToList();

//Overalldata.AddRange(temp1.Where(x=>x.TabClose=="Continue"));

Overalldata.AddRange(temp);

LOG1.Add(s.Value.BatchType, temp1);

Tab1.Add(s.Value.BatchType, temp);

closetiming.Add(s.Value.BatchType, temp3);

GetPingTime(s.Key);

TestTimetravel(gettime);

}

time1 = 120;

}

public SortedList<int, List<TestPingTime>> GetPingTime(int testid)

{

using (SqlConnection con = new SqlConnection(Properties.Settings.Default.Database))

{

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

List<TestPingTime> mytime = new List<TestPingTime>();

cmd.CommandText = "select max(CAST(LastPingtime AS datetime)) as mydate,SId from studtestpingtime where sid in (select Sid from TestTakenStatus where Status='Continue' and TestId='" + testid + "') and TestId='" + testid + "' group by SId ";

gettime.Clear();

SqlDataReader reader1;

reader1 = cmd.ExecuteReader();

while (reader1.Read())

{

mytime.Add(new TestPingTime { sid = reader1.GetInt32(1), pingTime = reader1.GetDateTime(0) });

}

gettime.Add(testid, mytime);

}

return gettime;

}

public void autolaunchtimer()

{

foreach (var auto in autolaunching)

{

if (auto.IsAutoLaunch == true)

{

var c = DateTime.Parse("4:15 pm");

var d = DateTime.Now;

var time = c.Subtract(d).TotalMinutes;

if (time == 0 || time > -3)

{

autolaunchdetails(auto.TestId, auto.BatchType);

}

}

}

}

public void TestTimetravel(SortedList<int, List<TestPingTime>> gettimeinfo)

{

foreach (var i in gettimeinfo)

{

List<TestPingTime> templist = new List<TestPingTime>();

templist.Clear();

if (i.Key != null)

{

foreach (var s in i.Value)

{

var dates = Convert.ToDateTime(changedateformat(s.pingTime.ToLocalTime()));

var r = DateTime.Now.Subtract(Convert.ToDateTime(dates)).TotalMinutes;

if (Math.Round(r) > 1)

{

var nms = istud.GetIndividualStudentDetail(s.sid);

templist.Add(new TestPingTime { name = nms[0].StudentName, sid = s.sid, pingTime = dates });

countlist.Add(new TestPingTime { name = nms[0].StudentName, sid = s.sid, pingTime = dates, testids = i.Key });

}

}

pingtimevisible.Add(i.Key, templist);

}

}

}

private void Testdetails(object sender, MouseButtonEventArgs e)

{

int present = 0;

int absents = 0;

int continues = 0;

Tagging = true;

//InitialDataload();

complete.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Ongo.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Border1.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Test.Background = new SolidColorBrush(Color.FromRgb(255, 56, 94));

pingborder.Visibility = Visibility.Collapsed;

absentvisible.Visibility = Visibility.Collapsed;

Studentgrid.Visibility = Visibility.Collapsed;

OnGoingGrid.Visibility = Visibility.Collapsed;

Testdatagrid.Visibility = Visibility.Visible;

absentvisible.Visibility = Visibility.Visible;

tabborder.Visibility = Visibility.Visible;

foreach (var s in currenttest.Values)

{

if (s.launched == false)

{

present += Overalldata.Where(x => x.Status == "Present" && x.testid == s.TestId).Count();

absents += Overalldata.Where(x => x.Status == "Absent" && x.testid == s.TestId).Count();

continues += Overalldata.Where(x => x.TabClose == "Continue" && x.testid == s.TestId).Count();

tabcount.Text = continues.ToString();

TestLogin.Text = present.ToString() + "/" + ActualStudents.ToString();

Copmletedstudents.Text = Overalldata.Where(x => x.AttendQus != "0" && x.TabClose == "Completed").Count().ToString() + "/" + present.ToString();

}

}

}

private void Backgroundwork\_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)

{

autoload();

g2.Visibility = Visibility.Collapsed;

Reload.Visibility = Visibility.Visible;

}

private void Backgroundwork\_DoWork(object sender, DoWorkEventArgs e)

{

preloaddata();

}

public int CountDecrement(int timeStarting)

{

if (timeStarting > 0)

{

return timeStarting - 1;

}

else

{

return 0;

}

}

private void Studentgrid\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

private void Logindata(object sender, RoutedEventArgs e)

{

complete.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Ongo.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Test.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Border1.Background = new SolidColorBrush(Color.FromRgb(249, 129, 58));

Testdatagrid.Visibility = Visibility.Collapsed;

Studentgrid1.Visibility = Visibility.Collapsed;

OnGoingGrid.Visibility = Visibility.Collapsed;

Studentgrid.Visibility = Visibility.Visible;

//logic

Studentgrid.ItemsSource = null;

if (Tagging == false)

{

Studentgrid.ItemsSource = LOG.OrderByDescending(x => x.Status == "Absent");

}

else

{

Studentgrid.ItemsSource = Overalldata.OrderByDescending(x => x.Status == "Absent");

}

}

private void CompletedClick(object sender, RoutedEventArgs e)

{

Studentgrid.UnselectAll();

Ongo.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Test.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

Border1.Background = new SolidColorBrush(Color.FromRgb(238, 238, 238));

complete.Background = new SolidColorBrush(Color.FromRgb(249, 129, 58));

Testdatagrid.Visibility = Visibility.Collapsed;

Studentgrid.Visibility = Visibility.Collapsed;

OnGoingGrid.Visibility = Visibility.Collapsed;

Studentgrid1.Visibility = Visibility.Visible;

//logic

Studentgrid1.ItemsSource = null;

if (Tagging == false)

{

Studentgrid1.ItemsSource = Tab.Where(x => x.Status == "Present").OrderByDescending(x => x.TabClose == "Continue");

}

else

{

Studentgrid1.ItemsSource = Overalldata.Where(x => x.Status == "Present").OrderByDescending(x => x.TabClose == "Continue");

}

}

private void AutoLaunch(object sender, RoutedEventArgs e)

{

var temp = Testdatagrid.SelectedCells;

var batcht1 = temp[0].Item as TestDetails;

var s = batcht1.StartTime.Add(TimeSpan.FromMinutes(-8));

using (SqlConnection con = new SqlConnection(Properties.Settings.Default.Database))

{

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "update TestDetails set IsAutoLaunch='" + true + "' where TestId='" + batcht1.TestId + "' and Batchtype='" + batcht1.BatchType + "'";

cmd.ExecuteNonQuery();

con.Close();

}

data();

}

public void data()

{

currenttest.Clear();

Testdatagrid.ItemsSource = null;

currenttest = icurrentdaytest.TestDetailsRepoistory(GetDateString(today));

Testdatagrid.ItemsSource = currenttest.Values;

foreach (var j in currenttest.Values)

{

if (j.IsAutoLaunch == true)

{

autolaunching.Add(new TestDetails { BatchType = j.BatchType, TestId = j.TestId, StartTime = j.StartTime, IsAutoLaunch = j.IsAutoLaunch });

}

}

preloaddata();

}

public void autolaunchdetails(int testid, string batch)

{

var bat = batch.Split('\_');

if (bat[0].ToUpper() == "UPCOMING")

{

using (SqlConnection con = new SqlConnection(Properties.Settings.Default.Database))

{

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "update TestDetails set BatchType='" + bat[1] + "' where TestId='" + testid + "'";

cmd.ExecuteNonQuery();

con.Close();

}

using (SqlConnection con = new SqlConnection(Properties.Settings.Default.Database))

{

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "update TestDetails set IsAutoLaunch='" + false + "' where TestId='" + testid + "' and Batchtype='" + bat[1] + "'";

cmd.ExecuteNonQuery();

con.Close();

}

data();

}

}

private void LaunchNow(object sender, RoutedEventArgs e)

{

l2.Visibility = Visibility.Visible;

var temp = Testdatagrid.SelectedCells;

MessageBox.Show("launching");

Button btn = sender as Button;

btn.Visibility = Visibility.Collapsed;

var batcht1 = temp[0].Item as TestDetails;

var bat = batcht1.BatchType.Split('\_');

if (bat[0].ToUpper() == "UPCOMING")

{

using (SqlConnection con = new SqlConnection(Properties.Settings.Default.Database))

{

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "update TestDetails set BatchType='" + bat[1] + "' where TestId='" + batcht1.TestId + "'";

cmd.ExecuteNonQuery();

con.Close();

}

using (SqlConnection con = new SqlConnection(Properties.Settings.Default.Database))

{

SqlCommand cmd = new SqlCommand();

cmd.Connection = con;

con.Open();

cmd.CommandText = "update TestDetails set IsAutoLaunch='" + false + "' where TestId='" + batcht1.TestId + "' and Batchtype='" + bat[1] + "'";

cmd.ExecuteNonQuery();

con.Close();

}

data();

}

l2.Visibility = Visibility.Collapsed;

}

private void Cancelclick(object sender, RoutedEventArgs e)

{

Warning.Visibility = Visibility.Collapsed;

}

private void Warningclick(object sender, RoutedEventArgs e)

{

Warning.Visibility = Visibility.Visible;

}

}

public static class MyExt

{

public static void PerformClick(this Button btn)

{

btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));

}

}

}