

## **Building Computer-Based Test (CBT) using MATLAB: Programming the Essential Types of Questions**

Baghdadi Ammar Awni Abbas\* and Mohammed Al-Mukhtar†  
University of Baghdad, Baghdad, IRAQ

### **Abstract**

MATLAB is considered one of the most important multipurpose programs. In this paper we propose a testing package that can create Computer-Based Tests (CBT). The package contains the most frequent question types that are adopted in the most prominent Learning Management Systems (LMS) such as Google Classroom, Moodle, Canvas, Blackboard, D2L, Joomla, Schoology, and Talent. Six types of questions are discussed: True or False questions (TorF), Multiple Choice Questions (MCQ) single choice and multiple choice, fill in the blank, essay questions, and matching. Each of these questions is built using the MATLAB App Designer tool that comes with MATLAB R2020a. The package uses an Excel spreadsheet as a storage for the exam's information, student's answers, and grades. The user can make an exam with an unlimited number of questions. The user can take an exam with two options, either without any help as a real test, or taking the exam as training, where a button to "show the correct answers" is visible. The grading of the exam is a mixed operation between the user and the computer, the fill in the blank and essay question are graded manually. All other forms of questions are graded automatically. The graphical user interface is built for English language use. The exams are static, and no form of adaptation is used. Testing the program showed that the results are 100% accurate for a specimen of 200 users undertaking 20 different exams.

**Keywords:** Computer-based test, learning management system, MATLAB, true or false questions, multiple choice questions, fill in the blank, essay questions.

### **1 Introduction**

There has been a gradual growth over the past 40 years. in CBT. as a suitable replacement for to paper and pencil testing. CBT was one of the most widely used internet uses in the late 1990s, but e-learning has recently gained significant importance, especially since the corona pandemic's emergence. Based upon reviewing the main LMS, six types of questions were selected to be built programmatically using the MATLAB App Designer.

The exam is a non-adaptive, fixed test where the question types and order are previously selected by the creator of the exam according to the type of materials examined. The user can navigate through the question using the next/previous button. The program has no time limit and the user must end the exam manually to get the final score and get the certificate to pass the exam. The program presented runs on a standalone computer. The user creates an exam and automatically the exam is stored in an Excel spreadsheet named after the exam name selected by the creator. The spreadsheet in the Excel file serves as an exam bank, where the creator sets a complete exam in every spreadsheet.

### **2 Related Work**

In [3] a comparison between CBT and paper-based exams for the postgraduate student is made. The majority of students preferred the first over the latter. Grading MCQ automatically without human interaction in [4], the program has 100% results accuracy. In [8] a fingerprint method is used to authenticate the examinee; the level of authentication is highly improved using these techniques. In [15] a biometric recognition method is used to authenticate exam entrance. In [16], methods and models of creating CBT are thoroughly explained, describing the ways to compare the models and describing the test delivery methods, finally; the validity issues are discussed; showing that it's a key issue when deciding the best model for the program. An application for CBT for smartphones (android) is presented in [14], the satisfaction for a sample of 30 students and a teacher is measured through a questionnaire, and a high rate of acceptance is noted. A novel approach to CBT is introduced in [12], where the examinee is asked to assemble objects on a computer screen; the test is made using Macro Media Flash. A comparison between traditional MCQ types of questions and innovative item formats in a CBT program was analyzed for IRT information with the three parameter and graded response models [9]. The waterfall model and the Reuse-oriented software process models are used in [2] to make a component-based software that recycles the same software element to make other components. A way to evaluate and minimize the length of a CBT on sentence comprehension is presented in [19], 5 to 8 minutes were reduced from the exam time with the same results. The analysis in [18] discusses the effectiveness of the bimasoft application as a medium for evaluating CBT learning using Android. Results of the study found that the application of the

\* College of Mass Media. Email: ammar.a@comc.uobaghdad.edu.iq

† Computer Center.

bimasoft application as a medium for evaluating CBT learning using Android was very good (84.64%). In [6] light is shed on the increase of electronic tools in education and learning, focusing on terminology, with diverse terms utilizing the same assessment approach within the literature, such as electronic assessment/evaluation, and online assessment/evaluation. The work [13] is the closest to our work, where a system to create/undertake exams is presented; the system is built using visual basic, HTML, and My SQL. The system proved to be very robust, stable, and error-free. The results of favoring computer-based testing complex "Profvybir" are presented in [10], 780 students took part in the assessment. This paper's method of occupational guidance can be used to implement the policy of the Ministry of Education and Science in the development of occupational guidance can be applied by a career advisor at school, while the CBT complex may be an additional component of the occupational training program. In [5] solutions to the current problems of CBT in Nigeria presented the user-friendly program using visual basic. The system is designed using the agile methodology through the extreme programming approach and unified modeling language was used to bring the view to real-life situations. Software is meant to be used for all kinds of CBT conducted or managed by universities in Nigeria. The top-to-down approach was adopted as the implementation approach for the development. The study of accessibility problems in CBT for blind persons is studied in [17] and gives some recommendations to facilitate the ease of use for them, the results showed that most CBT does not meet the expectations of visually impaired persons. In [1] there is discussion of the use of blended learning in Iraqi universities, where the proposed hybrid model has decreased the error rate from (0.00014) to (0.00013). An overview of the CBT models is made in [11]. Nigerian undergraduate students, participate in a CBT by comparing several modules studied by the students, the results of [7] showed that the students prefer CBT over paper-based exams.

## 2.1 Software Description

The main graphical user interface Figure 1 starts with a simple window that takes the user to three main parts of the program: 1-creating the exams 2-taking an exam that is made in the first part, either as a real exam or as a practice (which shows the correct answer button hidden according to the type of exam) 3-grading an exam taken in part two of the program. The fourth part of the main interface contains a brief description of the program and the programmer.

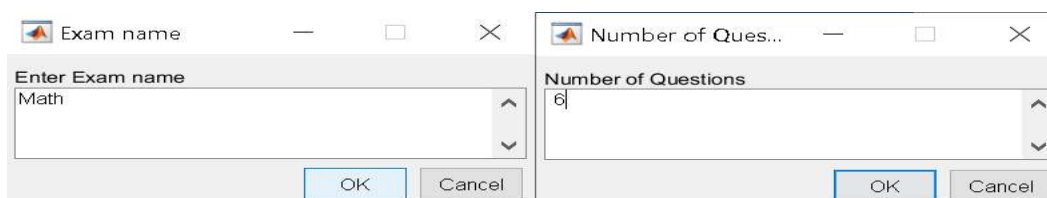


Figure 2: Getting the name of the new exam with the total amount of questions

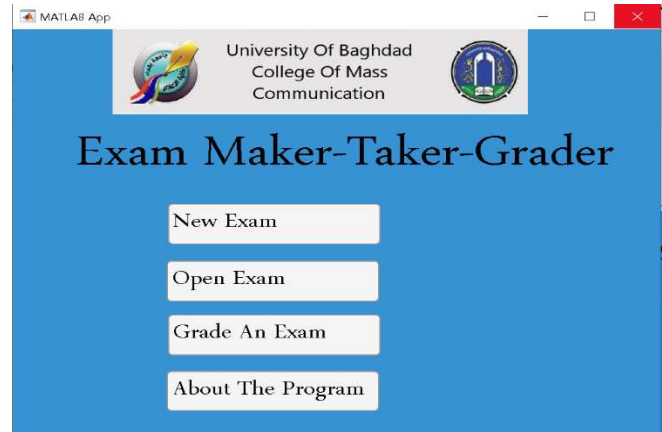


Figure1: Main graphical user interface

## 2.2 Part One: Creating a New Exam

The first part of the main GUI deals with creating a new exam, when this button is pressed, a message box popup asks the user to enter a proper name for the exam followed by another message box to specify the number of questions in the exam, as shown in Figure 2. The coding for these message boxes is illustrated in Code 1.

The first message will be turned into a spreadsheet in an Excel file that contains all the necessary information about the exam and the correct answers (including the total number of questions obtained from the second message box). Afterward, a dropdown menu appears asking the user to choose one of the six available types of questions as shown in Figure 3. The first row of the spreadsheet is reserved for the name of the exam cell (1,1) and cell (1,2), and the second row is also reserved for the number of the question in cells (2,1) and (2,2). Code 2 presents the line way to choose from the drop-down box.

There are six types of questions in the program:

1-True or False: The coding for this type of question in the Excel spreadsheet is made as followed: The first cell (x,1) is the code of the question which is given the number 1, and to the right a cell that contains the question cell(x,2) followed by a third cell for the correct answer (1 denotes a True answer and 2 denotes a False answer). The GUI for T or F question is shown in Figure 4. Code 3 presents the programming of the same question after retrieving the correct name of the spreadsheet, the total number of questions, and the current question pointer position.

```

prompt = {'Enter Exam name'};
dlgtitle = 'Exam name';
dims = [3 50];
definput = {' '};
b = inputdlg(prompt,dlgtitle,dims,definput);
nameoffile= "Name of Exam";
f = char(b);
writematrix(nameoffile, 'A.xls', "Sheet", f, "Range", 'A1')
writematrix(f, 'A.xls', "Sheet", f, "Range", 'B1')
setappdata(0, 'ExamName', f);
%Write Exam Name in the spreadsheet of the student answer
writematrix(nameoffile, 'A.xls', "Sheet", f, "Range", 'A2')
writematrix(f, 'A.xls', "Sheet", f, "Range", 'B2')
% Write Exam name in a seprate spreadsheet
writematrix(f, 'A.xls', 'WriteMode', "append")
% Save the name of the exams
writematrix(f, 'A.xls', "Sheet", f, "Range", 'B1')
%Enter Number of Question
prompt = {'Number of Questions'};
dlgtitle = 'Number of Questions';
dims = [3 50];
definput = {' '};
QuestionNo = inputdlg(prompt,dlgtitle,dims,definput);
g = char(QuestionNo);
writematrix("Number of Questions", 'A.xls', "Sheet", f, "Range", 'A2')
writematrix(g, 'A.xls', "Sheet", f, "Range", 'B2')
setappdata(0, 'NumberQuestions', g);

```

Code 1 Coding the message box and changing the input to a spreadsheet

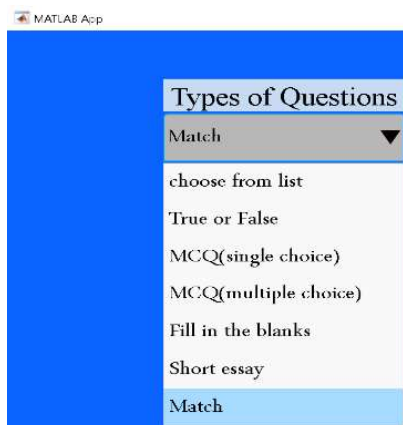


Figure 3: Question types

```

value = app.DropDown.Value ;
switch value
case 'True or False'
run Threetruefalsefinal.mlapp
case 'MCQ(single choice)'
run FourMCQS.mlapp
case 'MCQ(multiple choice) '
run FiveMCQM.mlapp
case 'Fill in the blanks'
run SixFillBlank.mlapp
case 'Short essay'
run SevenEssay.mlapp
otherwise
run EightMatch.mlapp
end

```

Code 2: Choosing from drop down menu items

After typing the question and choosing the correct answer the creator must push the save button and wait for the confirmation message box “Your Question is Saved” before choosing to return to the main page to pick another form of question or staying with the same type of question.

2-Multiple Choice Questions (a single Choice from four choices): The coding for this type of question in the Excel spreadsheet is made as follows: The first cell (x,1) is the code of the question which is given the number 2, to the right a cell that contains the question cell(x,2) followed by a third cell (x,3) for the correct answer (1 denotes a first answer is correct, 2 denotes a second answer is correct, 3 denotes a second answer is correct and 4 denotes the fourth answer is the right answer). The cells(x,4) to (x,7) contain the four choices for the question. This type of question is

made using a radio button which enables only one choice for each question. The GUI for MCQ single-choice question is shown in Figure 5, while the coding for such a question is presented in Code 4.

3-Multiple Choice Questions (with multiple choices): This type of question uses checkboxes instead of radio buttons which allows the user to select more than one option. The coding for this type of question in the Excel spreadsheet is made as follows: The first cell (x,1) is the code of the question which is given the number 3, to the right is a cell that contains the question cell(x,2) followed by a third cell (x,3) for the correct answer (see Table 1 for the different combinations of the answer). The cells(x,4) to (x,7) contain the four choices for the question. The MCQ multiple choices question is shown in Figure 6.

The MCQ (multiple choice) has almost the same GUI as the MCQ (single choice) with the same save and back to questions GUI buttons. The coding for that type of question is presented in Code 5.

4-Fill in the blank: The coding for this type of question in the Excel spreadsheet is made as followed: The first cell (x,1) is the code of the question which is given the number 4, and to the right is a cell that contains the question cell(x,2) followed by cell (x,3) to (x,6) for the correct answers. This type of question is graded manually unlike the first three types of questions which are graded automatically by the program. The fill in the blank question is shown in Figure 7. The coding for fill in the blank is presented in Code 6.

5-Essay Questions: The student must write the answer in the edit text field. The coding for this type of question in the Excel spreadsheet is made as followed: The first cell (x,1) is the code of the question which is given the number 5, to

the right is a cell that contains the question cell(x,2) followed by cell (x,3) for the correct answers. This type of question is also graded manually like the fill in the blank question. Essay GUI is shown in Figure 8. The coding for the essay question is presented in Code 7.

6-Match Questions: The student is asked to match the options on the right to the ones on the left; the matching is made by clicking the number that carries the first sentence on the right first then clicking the matching sentence on the left. The user is asked to enter the correct option on the left that match the sentence on the right and this was considered the correct answer. In a real exam both the left and right columns of options are distributed randomly each time, an exam is opened (as a real exam or as a training exam). A click on the number of the sentence will change the color of the number and if it was followed by clicking the matching option on the left, the same color will appear on the left. The coding for this type of question in the Excel

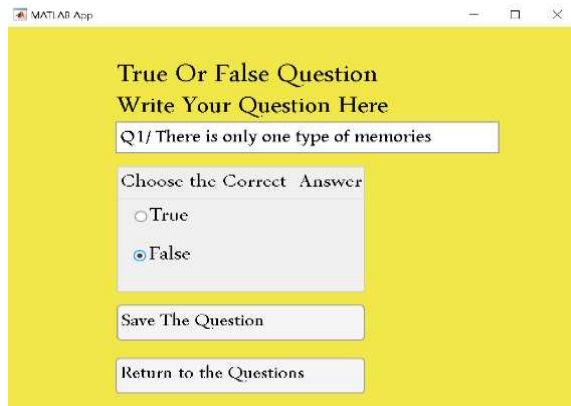


Figure 4: True or false question type

```
n=n+1 %Question counter
o=o+1; %Cell Counter
o1=num2str(o);
o2='A';
o3='B';
o4=append(o2,o1)
o5=append(o3,o1)
o6='C';
o7=append(o6,o1);
writematrix(1,'A.xls','Sheet',m,"Range",o4)
writematrix(app.EditField.Value,'A.xls','Sheet',m,"Range",o5)
setappdata(0,'questioncounter',n);
setappdata(0,'CellCounter',o);
%Buttons Information
mm=1 ;
%False Choice
mm=app.FalseButton.Value;
if mm==1
writematrix(2,'A.xls','Sheet',m,"Range",o7)
elseif mm==0
writematrix(1,'A.xls','Sheet',m,"Range",o7)
b=msgbox('Your Question is Saved' 'Your Question is Saved' )
end
```

Code 3: The T or F programming

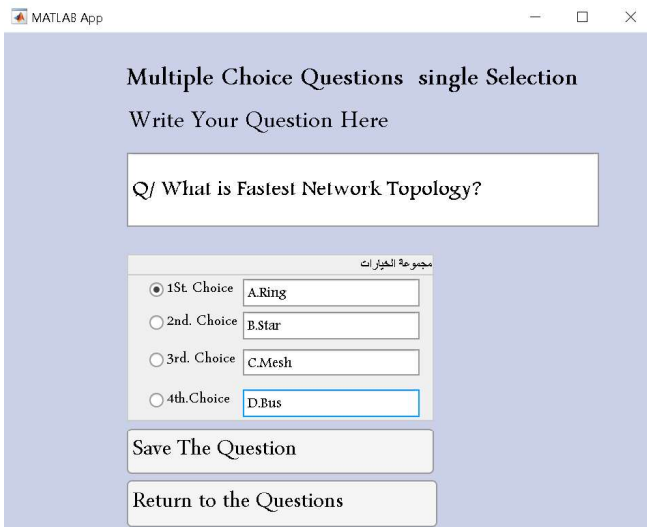


Figure 5: Multiple choice questions (a single choice from four choices)

```
%The right choice in cell 7
if app.StChoiceButton.Value==1 %The Answer is the first choice
mmm=1
writematrix(1,'A.xls','Sheet',m,"Range",o7)
elseif app.ndChoiceButton.Value==1 %The answer is the second choic
mmm=2
writematrix(2,'A.xls','Sheet',m,"Range",o7)
elseif app.rdChoiceButton.Value==1 %The answer is the third
mmm=3
writematrix(3,'A.xls','Sheet',m,"Range",o7)
else
mmm=4% the answer ir fourth choice
writematrix(4,'A.xls','Sheet',m,"Range",o7)
end
% Placing the ansers in cells 9 10 11 12
%1st answer
o8='D';
o9=append(o8,o1)
writematrix(app.EditField_2.Value,'A.xls','Sheet',m,"Range",o9)
%2nd answer
o10='E';
o11=append(o10,o1)
writematrix(app.EditField_3.Value,'A.xls','Sheet',m,"Range",o11)
%3rd answer
o12='F';
o13=append(o12,o1)
writematrix(app.EditField_4.Value,'A.xls','Sheet',m,"Range",o13)
%4th answer
o14='G';
o15=append(o14,o1)
writematrix(app.EditField_5.Value,'A.xls','Sheet',m,"Range",o15)
else
f=msgbox('You reached the Maximum Number of Questions')
end
```

Code 4: Programming MCQS

Table 1: The different possible combinations of MCQ (multiple choice) answer

| Case No. | CheckBox1 | CheckBox2 | CheckBox3 | CheckBox4 | Answer No. |
|----------|-----------|-----------|-----------|-----------|------------|
| 1        | N         | N         | N         | N         | 1          |
| 2        | Y         | N         | N         | N         | 2          |
| 3        | N         | Y         | N         | N         | 3          |
| 4        | Y         | Y         | N         | N         | 4          |
| 5        | N         | N         | Y         | N         | 5          |
| 6        | Y         | N         | Y         | N         | 6          |
| 7        | N         | Y         | Y         | N         | 7          |
| 8        | Y         | Y         | Y         | N         | 8          |
| 9        | N         | N         | N         | Y         | 9          |
| 10       | Y         | N         | N         | Y         | 10         |
| 11       | N         | Y         | N         | Y         | 11         |
| 12       | Y         | Y         | N         | Y         | 12         |
| 13       | N         | N         | Y         | Y         | 13         |
| 14       | Y         | N         | Y         | Y         | 14         |
| 15       | N         | Y         | Y         | Y         | 15         |
| 16       | Y         | Y         | Y         | Y         | 16         |

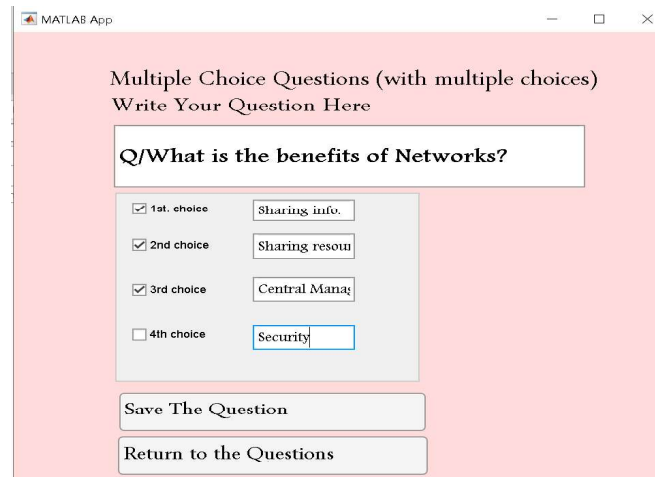


Figure 6: Multiple choice questions (with multiple choices)

```

%Buttons Info|
state=0;
if (app.stchoiceCheckBox.Value==0)&&(app.ndchoiceCheckBox.Value==0)&&(app.rdchoiceCheckBox.Value==0)&&(app.thchoiceCheckBox.Value==0)
    state=1
    writematrix(1,'A.xls','Sheet',m,"Range",o7)
elseif (app.stchoiceCheckBox.Value==1)&&(app.ndchoiceCheckBox.Value==0)&&(app.rdchoiceCheckBox.Value==0)&&(app.thchoiceCheckBox.Value==0)
    state=2
    writematrix(2,'A.xls','Sheet',m,"Range",o7)
elseif (app.stchoiceCheckBox.Value==0)&&(app.ndchoiceCheckBox.Value==1)&&(app.rdchoiceCheckBox.Value==0)&&(app.thchoiceCheckBox.Value==0)
    state=3
    writematrix(3,'A.xls','Sheet',m,"Range",o7)
elseif (app.stchoiceCheckBox.Value==1)&&(app.ndchoiceCheckBox.Value==1)&&(app.rdchoiceCheckBox.Value==0)&&(app.thchoiceCheckBox.Value==0)
    state=4
    writematrix(4,'A.xls','Sheet',m,"Range",o7)
elseif (app.stchoiceCheckBox.Value==0)&&(app.ndchoiceCheckBox.Value==0)&&(app.rdchoiceCheckBox.Value==1)&&(app.thchoiceCheckBox.Value==0)
    :
    :
    :
elseif (app.stchoiceCheckBox.Value==1)&&(app.ndchoiceCheckBox.Value==0)&&(app.rdchoiceCheckBox.Value==1)&&(app.thchoiceCheckBox.Value==0)
    state=15
    writematrix(15,'A.xls','Sheet',m,"Range",o7)
else
    state=16
    writematrix(16,'A.xls','Sheet',m,"Range",o7)

```

Code 5: MCQ (multiple choices)

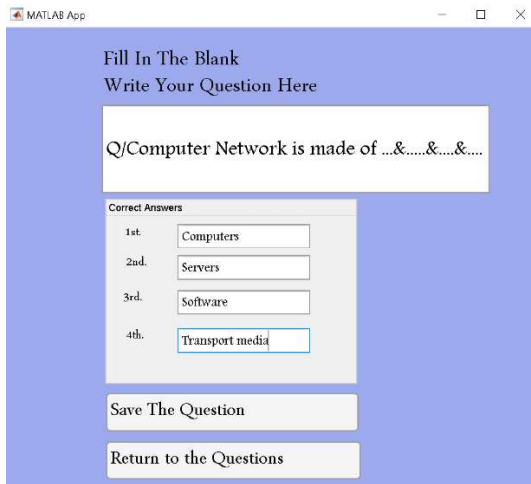


Figure 7: Fill in the blank Questions.

```

        writematrix(4, 'A.xls', "Sheet", m, "Range", o4) %choosing Q. Type
writematrix(app.EditField.Value, 'A.xls', "Sheet", m, "Range", o5) %Writing the Q.
        setappdata(0, 'questioncounter', n);
        setappdata(0, 'CellCounter', o);
o8='D'; %1st Answer
o9=append(o8, o1)
writematrix(app.EditField_2.Value, 'A.xls', "Sheet", m, "Range", o9)
o10='E'; %2nd Answer
o11=append(o10, o1)
writematrix(app.EditField_3.Value, 'A.xls', "Sheet", m, "Range", o11)
o12='F'; %3rd Answer
o13=append(o12, o1)
writematrix(app.EditField_4.Value, 'A.xls', "Sheet", m, "Range", o13)
o14='G'; %4th Answer
o15=append(o14, o1)
writematrix(app.EditField_5.Value, 'A.xls', "Sheet", m, "Range", o15)
else
    f = msgbox('You reached the Maximum Number of Questions')
end
end

```

Code 6: Fill in the blank

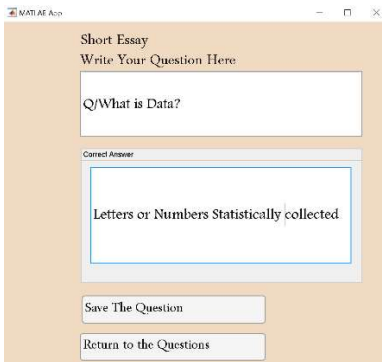


Figure 8: Essay question

```

writematrix(5, 'A.xls', "Sheet", m, "Range", o4) %Choosing Q. Type
%Writing Question
writematrix(app.EditField.Value, 'A.xls', "Sheet", m, "Range", o5)
        setappdata(0, 'questioncounter', n);
        setappdata(0, 'CellCounter', o);
        o8='C'; %Question Answer
o9=append(o8, o1)
writematrix(app.EditField_2.Value, 'A.xls', "Sheet", m, "Range", o9)
else
    f = msgbox('You reached the Maximum Number of Questions')
end
end

```

Code 7: Essay questions.

spreadsheet is made as followed: The first cell (x,1) is the code of the question which is given the number 6, to the right a cell(x,2) to cell(x,5) is the sentence choices on the right, afterward the cells(x,6) to (x,9) contains their corresponding matching options on the left. This type of question is graded automatically. Match GUI is shown in Figure 9. The coding specifies the question type and gets the different options in the two columns presented in Code 8.

The Excel spreadsheet that will be generated corresponds to an exam named 'computer science' with 6 questions (one for each type of question) with questions stated in Figures 4-9 presented in Figure 10.

### 2.3 Part Two: Opening an Existing Exam

The program can make a test for the user either for making a real CBT or for practicing with exams that are in the program's bank of questions. After clicking the opening an existing test is pressed, and a message box appears asking for the name of the user followed by another message box asking the user to specify that the test is a real CBT or a practice test, as shown in Figure 11. Coding such a task is presented in code 9.

The second message box input will have an impact on showing the button of 'show the correct answer', which will be invisible to the examinee taking a real CBT. Afterward, the user is asked to select a test from the existing available exams. The spreadsheet saved in the first part is turned into components in a list using the (listdlg) function. As illustrated in Figure 12 the coding for such a list is presented in Code 10.

Figure 13 shows the difference between a real CBT (a), and a practice test for the same exam (b), pressing the correct button will highlight the correct button in the autocorrect questions. (c) and (d). Coding (a) and (b) are presented in Code 11 (a), while the code for c is indicated in Code 11 (b).

The manually corrected questions (fill in the blank and essay) presented in Figure 14, and the "show the correct answer button" pressed and the questions with the correct answers are retrieved from the question spreadsheet and placed in the correct position.

The matching question will display the left and right sides of options randomly each time this type of question is invoked by an exam; with the correct match saved in the original exam. A click on the right side will cause a change in the color, and when the matching option on the left is clicked the same color appears and so on. Figure 15 shows the match question in a real exam.

Like creating a new exam, answering an exam (as a real exam or as practice) will create a new spreadsheet every time the program is used. The coding for the various question is given in Table 2 and Table 3.

The answers are coded as shown in Table 3.

A spreadsheet is generated containing the student answers in Figure 16. This sheet is given the name of the examinee and contains the name of the exam taken as well as the different answers from the examinee's response to the different questions.

## 2.4 Part Three: Grading an Exam

This part is responsible for giving the proper grades to the examinee, the grading is semiautomatic. The fill in the blank and the essay questions are graded manually while the program automatically grades the other types. The pressing of the grade an exam button will display a list of the available answers, as shown in Figure 17.

After choosing the desired exam, the questions appear in the order that they were saved with the examinee's answer plus the

Figure 9: Match questions

```
writematrix(6, 'A.xls', "Sheet", m, "Range", o4) %choosing Q.Type
    setappdata(0, 'questioncounter', n);
    setappdata(0, 'CellCounter', o);
writematrix(app.EditField.Value, 'A.xls', "Sheet", m, "Range", o5) %1st answer
writematrix(app.EditField_2.Value, 'A.xls', "Sheet", m, "Range", o7) %2nd answer
o8='D'; %3rd answer
o9=append(o8, o1)
writematrix(app.EditField_3.Value, 'A.xls', "Sheet", m, "Range", o9)
o10='E'; %4th answer
o11=append(o10, o1)
writematrix(app.EditField_4.Value, 'A.xls', "Sheet", m, "Range", o11)
o12='F'; %1st solution
o13=append(o12, o1)
writematrix(app.EditField_5.Value, 'A.xls', "Sheet", m, "Range", o13)
o14='G'; %2nd solution
o15=append(o14, o1)
writematrix(app.EditField_6.Value, 'A.xls', "Sheet", m, "Range", o15)
o16='H'; %3rd solution
o17=append(o16, o1)
writematrix(app.EditField_7.Value, 'A.xls', "Sheet", m, "Range", o17)
o18='I'; %4th solution
o19=append(o18, o1)
writematrix(app.EditField_8.Value, 'A.xls', "Sheet", m, "Range", o19)
```

Code 8: Match question

|   | A   | B                                       | C | D              | E               | F              | G        | H      | I    |
|---|---|---|---|----------------|-----------------|----------------|----------|--------|------|
| 1 | Name of Exam                                  | Computer Science                        |   |                |                 |                |          |        |      |
| 2 | Number of Questions                           | 6                                       |   |                |                 |                |          |        |      |
| 3 | 1 Q/There is only one type of Memories?       |   |   | 2              |                 |                |          |        |      |
| 4 | 2 Q/What is the fastest network topology?     |   |   | 1 Ring         | Star            | Mesh           | Bus      |        |      |
| 5 | 3 Q/What is the benefits of networks?         |   |   | 8 Sharing Info | SharingResource | Central Manage | Security |        |      |
| 6 | 4 Q/Computer network is made.....&.....&..... |   |   | Computers      | Servers         | Software       | Cables   |        |      |
| 7 | 5 Q/What is Data?                             | It is a collection of Numbers & Letters |   |                |                 |                |          |        |      |
| 8 | 6 Input Device                                | Output Device                           |   | Network Device | Topolgy         | Keyboard       | Monitor  | Router | Star |

Figure 10: The generated spreadsheet after creating an exam

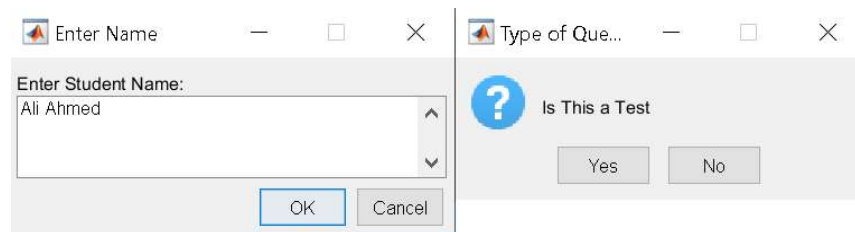


Figure 11: Name of the examiner and type of test message boxes

```

%writing the new student name
prompt = {'Enter Student Name:'};
dlgtitle='Enter Name';
dims = [3 50];
definput = {' '};
bo = inputdlg(prompt,dlgtitle,dims,definput);
nameofstudent= "Name of $student";
ro = char(bo);
writematrix(nameofstudent,'B.xls',"Sheet",ro,"Range",'A1')
writematrix(ro,'B.xls',"Sheet",ro,"Range",'B1')
setappdata(0,'StudentName',ro);
%writing the name of the new student
writematrix(ro,'B.xls','WriteMode',"append")
%Determine if this is an exam or Practise
flagexamOtrain = questdlg('Is This a Test', ...
    'Type of Questions ', ...
    'Yes','No','');
switch flagexamOtrain
    case 'Yes'
        bb=1
        setappdata(0,'ExamOTrain',bb);
    otherwise
        bb=2
        setappdata(0,'ExamOTrain',bb);
end

```

Code 9: Dialog boxes



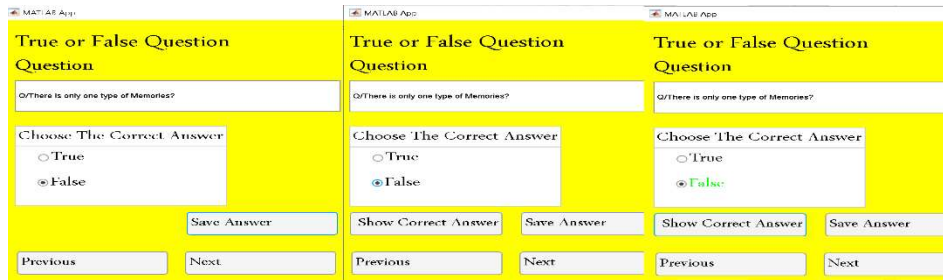


```

exam= readcell('A.xls');%change choice to a spreadsheet
exam1=char(exam);
[indx,tf] = listdlg('ListString',exam1);
n=indx;
n1=exam(n);
n2=char(n1);
writematrix('name of exam','B.xls',"Sheet",ro,"Range",'A2')%save Exam Name
writematrix(n2,'B.xls',"Sheet",ro,"Range",'B2')
n3 = readcell('A.xls','Sheet',n2);%Reading the exam
setappdata(0,'sheetname',n2)
    
```

Figure 12: Available exams list

Code 10: The dialog box for choosing an exam



(a)

(b)

(c)

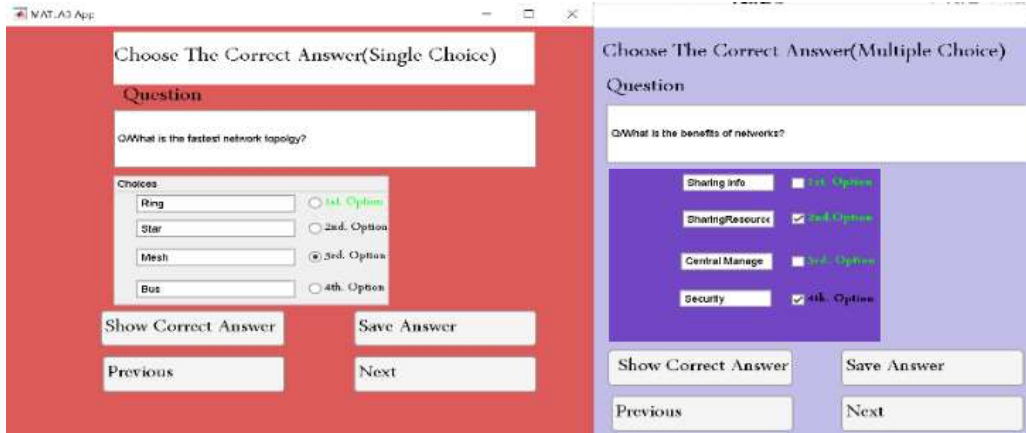


Figure 13: Real & practice tests (a) (b)

```

%Showing/Unshowing the correct answer button
k= getappdata(0,'Exam0Train')
switch k
case 1
    app.ShowCorrectAnswerButton.Visible='off'
otherwise
    app.ShowCorrectAnswerButton.Visible='on'
end

%Showing the Correct Answer in Green
A1= getappdata(0,'sheetname')
n3 = readcell('A.xls','Sheet',A1);
A2=getappdata(0,'lowerbounder')
jj=n3(A2,3);
jj1=cell2mat(jj);
if jj1==1
    app.TrueButton.FontColor= [0,1,0]
elseif jj1==2
    app.FalseButton.FontColor= [0,1, 0]
end
    
```

Code 11: T or F question with the ability to hide show the right answer

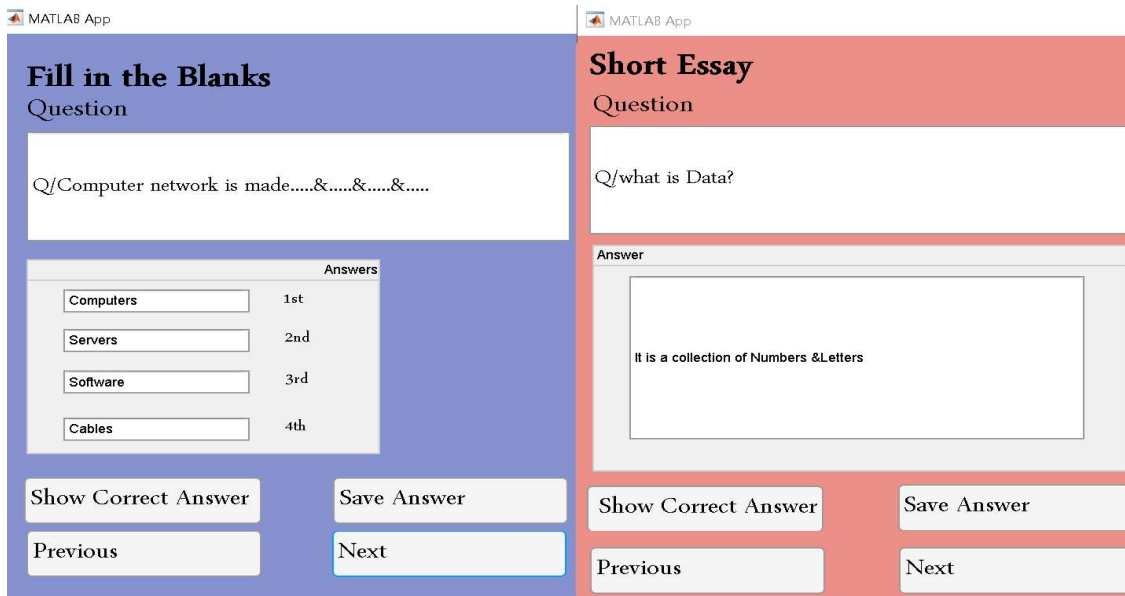


Figure 14: Showing the correct answer in manually corrected questions

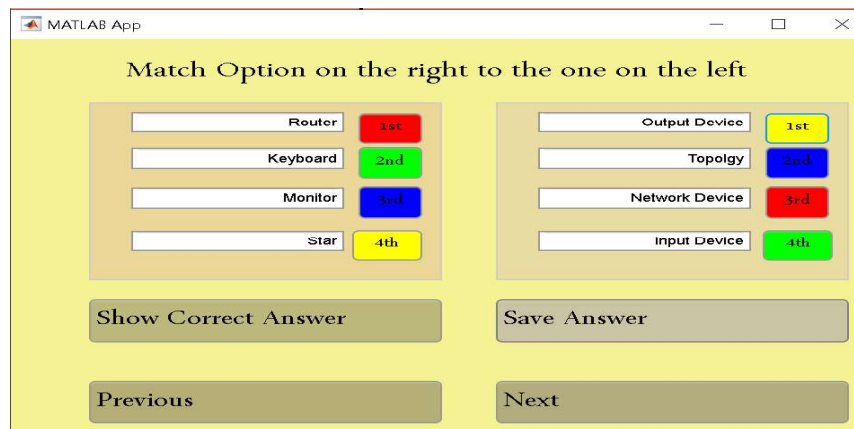


Figure 15: The Matching question in an exam

Table 2: The header for the answer

| Cell name | contents                                   |    |   |
|-----------|--|----|---|
| A1        | 'Name of student'                          | B1 | Actual name of the student  |
| A2        | 'Name of exam'                             | B2 | Actual name of the exam   |
| A3.....An | Type of Question by number(1,2,3,4,5 or 6) | B3 | The student's answer (might be True or False according to Table 3 |

Table 3: Coding the answers

| Cell name | contents                         | Cell name             | contents   |   |
|-----------|----------------------------------|-----------------------|--|---|
| Ax        | 1:(for T or F question)          | Bx                    | 1 or 2   | 1: True<br>2: False   |
| Ax+1      | 2 :(MCQ single choice)           | Bx+1                  | 1,2,3 or 4   | 1:1 <sup>st</sup> . answer selected<br>2:2nd. Answer<br>3:3 <sup>rd</sup> answer<br>4:4 <sup>th</sup> .answer |
| Ax+2      | 3 :(MCQ multiple choice)         | Bx+2                  | From 1 to 16   | According to the state illustrated in Table (1)   |
| Ax+3      | 4 :( Fill in the blank question) | Bx+3,Cx+3,Dx+3, Ex+3  | Answers from 1 to 4 according to the question                  |   |
| Ax+4      | 5:( Essay question)              | Bx+4                  | Answer to the question   |   |
| Ax+5      | 6 :(Match question)              | Bx+5,Cx+5,Dx+5, Ex+5, | The select items on the right are coded in numbers from 1 to 4 |   |
|           |                                  | Fx+5,Gx+5, Hx+5,Ix+5  | The select items on the left are coded in numbers from 1 to 4  |   |

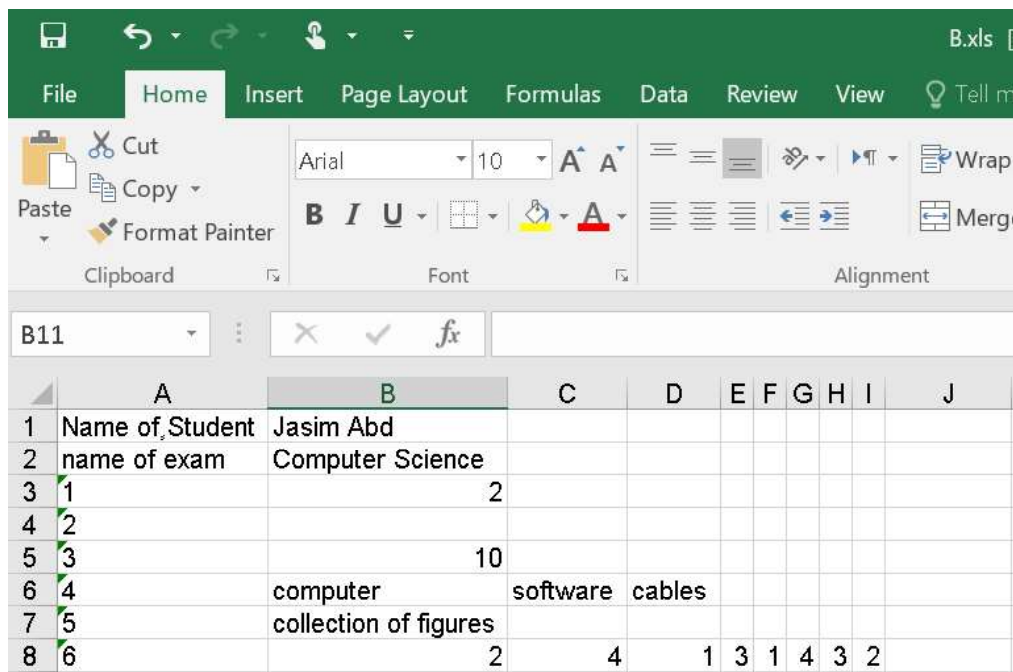


Figure 16: The spreadsheet was generated from the examinee’s answer

correct answer (for the autocorrect questions). The left side contains the name of the student, exam name, total score points, and score for an individual question (a spinner component that has a default number set to the grade number for a question. This number can be changed by the exam corrector if he wants to change the mark for the auto-graded question or give the proper degree for the fill in the blank or essay.

After giving the proper degree “Degree accreditation button” is pressed, and the total degree is calculated, as shown in Figure 18.

For the manual grading questions, the spinner component is used to select the proper mark for the question. When the final question is reached and this button is pressed followed by the “Degree accreditation button” the final mark and the name of

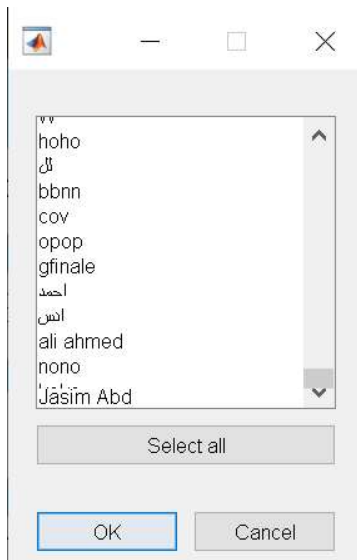


Figure 17: List of the available examinee

the student is stored in a third spreadsheet.

### 3 Conclusion

Information and communication technology (ICT) advancements will improve community life and allow governments to operate more efficiently and sustainably. The CBT is one of the cornerstones of any LMS. From the proposed system, we can draw the following conclusions:

- 1-The three essential steps for the system (creating, taking, and grading) are made by MATLAB in conjunction with Microsoft Excel so that each question can be traced thoroughly and clearly in all three steps no matter the size of the exam. The debugging was made for each type of question separately with no margin of error.
- 2-There are six types of questions in the program chosen according to the frequency of appearance in the major LMS and CBT programs. Many other common types of questions like re-order and Hot spot can be easily added to the program.
- 3- Many new types of questions can be invented and added to the program, exploiting MATLAB capabilities. For example, MATLAB Simulink is one of the most promising tools to be used for a real-time simulation in all fields of technology and creating questions about these situations.
- 4-Although the exams are predefined(static) because the program is mainly designed to be used in real exams; each participating student has equal opportunity questions, further analysis of the answers can be made, to measure the response patterns and use that to tweak the questions. This topic is worth further investigation and can be used in future work.
- 5-The program interfaces are designed in the English language and they can be easily changed to other left-to-right languages simply by changing the interfaces for each question. On the other hand, right-to-left needs an added effort to change the alignment of the text boxes to use these languages in the program.

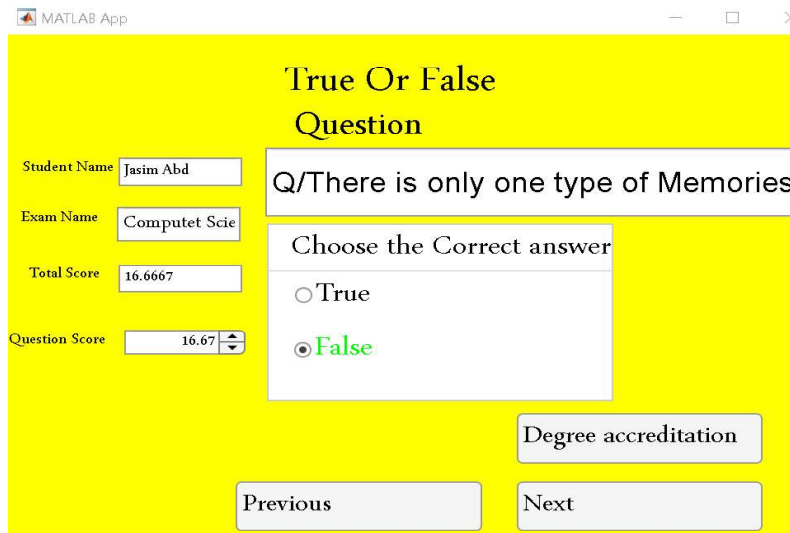


Figure 18: T or F question grading

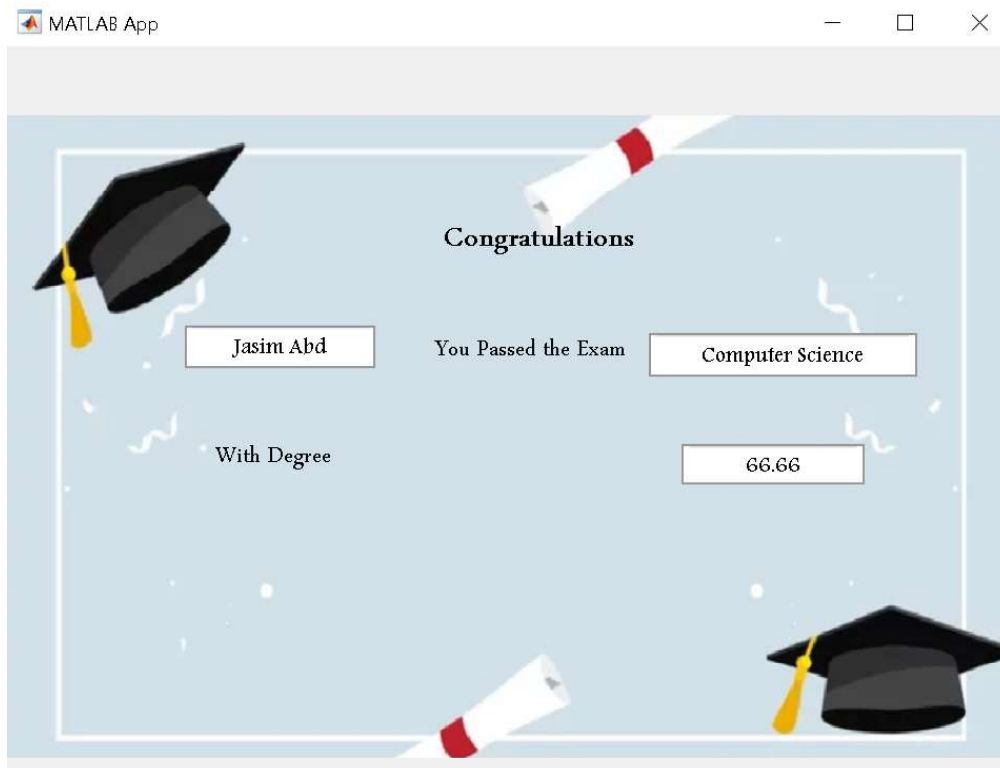


Figure 19: Pass exam certification

The image shows a screenshot of an Excel spreadsheet in Compatibility Mode. The spreadsheet contains a table with the following data:

|   | A            | B                | C      | D      | E | F | G |
|---|--------------|------------------|--------|--------|---|---|---|
| 1 | Student Name | Exam Name        | Degree | Serial |   |   |   |
| 2 | ali ahmed    | jrkdhj vrlm      | 75     | 2      |   |   |   |
| 3 | Jasim abd    | Computer tech.   | 54     | 3      |   |   |   |
| 4 | Ali mohammed | computer science | 65     | 4      |   |   |   |
| 5 |              |                  |        |        |   |   |   |

Figure 20: Record of examinee

**References**

[1] R. Abdulhussien and H. Najeeb, "Improving Measurement of Effectiveness of Blended Learning in Iraqi Education Using SVM," *Iraqi Journal of Science*, 63(9):4057-4066, 2022.

[2] M. Ajinaja, "The Design and Implementation of a Computer Based Testing System Using Component-Based Software Engineering," *International Journal of Computer Science and Technology*, 8(9):58-65, March 2017.

[3] A. Baghdadi, "The Computer Based Tests: A Digital

- Substitution for the Iraqi Postgraduate Students,” *Ibn Al-Haitham Jour. for Pure & Appl. Sci.*, 31(3):171-177, 2018, <https://doi.org/10.30526/31.3.2019>.
- [4] A. Baghdadi, “An Automatic System to Grade Multiple Choice Questions Paper-Based Test.” *J. of Al-Anbar University for Pure Science*, 3(1):174-181, 2009.
- [5] O. Chibuzo and D. Isiaka, “Design and Implementation of Secure Browser for Computer-Based Tests,” *International Journal of Innovative Science and Research Technology*, 5(8):1347-1356, August 2020.
- [6] N. Dogan, N. UYSAL, and R. Hambelton, “An Overview of E-Assessment Hacettepe University Journal of Education,” *University Journal of Education*, 35(Special Issue):1-5, 2020, doi: 10.16986/HUJE.2020063669.
- [7] S. Ejim, “An Overview of Computer Based Test”, DOI: 10.13140/RG.2.2.32040.88326, February 2017.
- [8] A. Ewwiekpaeffe and V. Eyinla, “Implementing Fingerprint Authentication in Computer-Based Tests,” *Nigerian Journal of Technology*, 40(2):284-291, 2021.
- [9] M. Jodin, “Measurement Efficiency of Innovative Item Formats in Computer-Based Testing,” *Journal of Educational Measurement*, 40(1):1-1s, 2003.
- [10] O. Kravchenko, N. Shelenkova, M. Shelenkova, and I. Boichevskaya, “Computer-Based Testing Complex “Profvybir”: Occupational Guidance Diagnostics Journal of Physics: Conference Series, Volume 1828, 2020 International Symposium on Automation, Information and Computing (ISAIC 2020) 2-4 December 2020, Beijing, China. doi:10.1088/1742-6596/1828/1/012125.
- [11] R. Luecht and S. Sireci S “A Review of Models for Computer-Based Testing,” College Board Research Report, 2011-12.
- [12] K. Maneekhao N. Jaturapitakkul R. Watson R., and S. Tepsuriwong., “Developing an Innovative Computer-Based Test,” *Prospect*, 21(2):34-46, August 2006.
- [13] Naseef Husam Mohammad, Nada Thanoon Ahmed, and Yasmin Makki Mohialden, “Development of Multiple Computer-Based Testing System Using Open-Source Programming Model,” *Journal of Physics: Conference Series* 1804 012063, 2021, 012063 IOP Publishing doi:10.1088/1742-6596/1804/1/012063.
- [14] H. Nurhikmah, H. Abdul Gani, M. Pratama, and H. Wijaya, “Development of an Android-Based Computer Based Test (CBT) In Middle School,” *Journal of Education Technology*, 5(2):272-281, 2021.
- [15] E. Nweneke, “A Secure Online Computer-Based Test System Using Facial Recognition Biometric Authentication,” A Case Study of Mountain Top University, 2021.
- [16] F. Okocha, “Student Perception of Computer-Based Testing in Kwara State, Nigeria,” *International Journal of Web-Based Learning and Teaching Technologies*, 17(1):1-11, 2022. DOI: 10.4018/IJWLTT.294575.
- [17] P Patel and A. Karkare, “Accessibility Evaluation of Computer Based Tests,” arXiv:1905.01825v1, 2019.
- [18] M. Umar and A. Jaya, “The Bimasoft Application as Computer Based Test (CBT) Learning Evaluation Media: An Analysis of the Effectiveness Using Android,” *Jurnal Pengkajian Ilmu dan Pembelajaran Matematika dan IPA IKIP Mataram*, 10:821-830, 3 July 2022.
- [19] M. Schurig, J. Jungjohann, and M. Gebhardt2, “Minimization of a Short Computer-Based Test,” *Reading Frontiers in Education*, 6:1-12 | Article 684595 2021.



**Baghdadi Ammar Awni Abbas** is an Assistant Professor of information technology. He received his Ph.D. from Kharkiv in Radioelectronics/Ukraine in 2015. He has over 30 published papers in 3 languages.

He is now a member in the Faculty of the Department of Radio and Television/College of mass media/University of Baghdad. His lifetime devotion is programming and teaching others how to program. MATLAB and C language are his favorite tools where he used these tools in image processing, computer vision pattern recognition, human-computer interaction, and artificial intelligence. He is currently working on a large project of interactive lectures for undergraduate students.



**Mohammed Al-Mukhtar** received his M.Sc. degree in Information and Communication Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019. Currently, he is working as a Lecturer at the University of Baghdad. His research interests include Computer Application, Medical Image Analysis

with deep learning techniques, Computer Vision, Object Recognition, and Object Segmentation.