Title: MedQuiz – a medical quiz platform

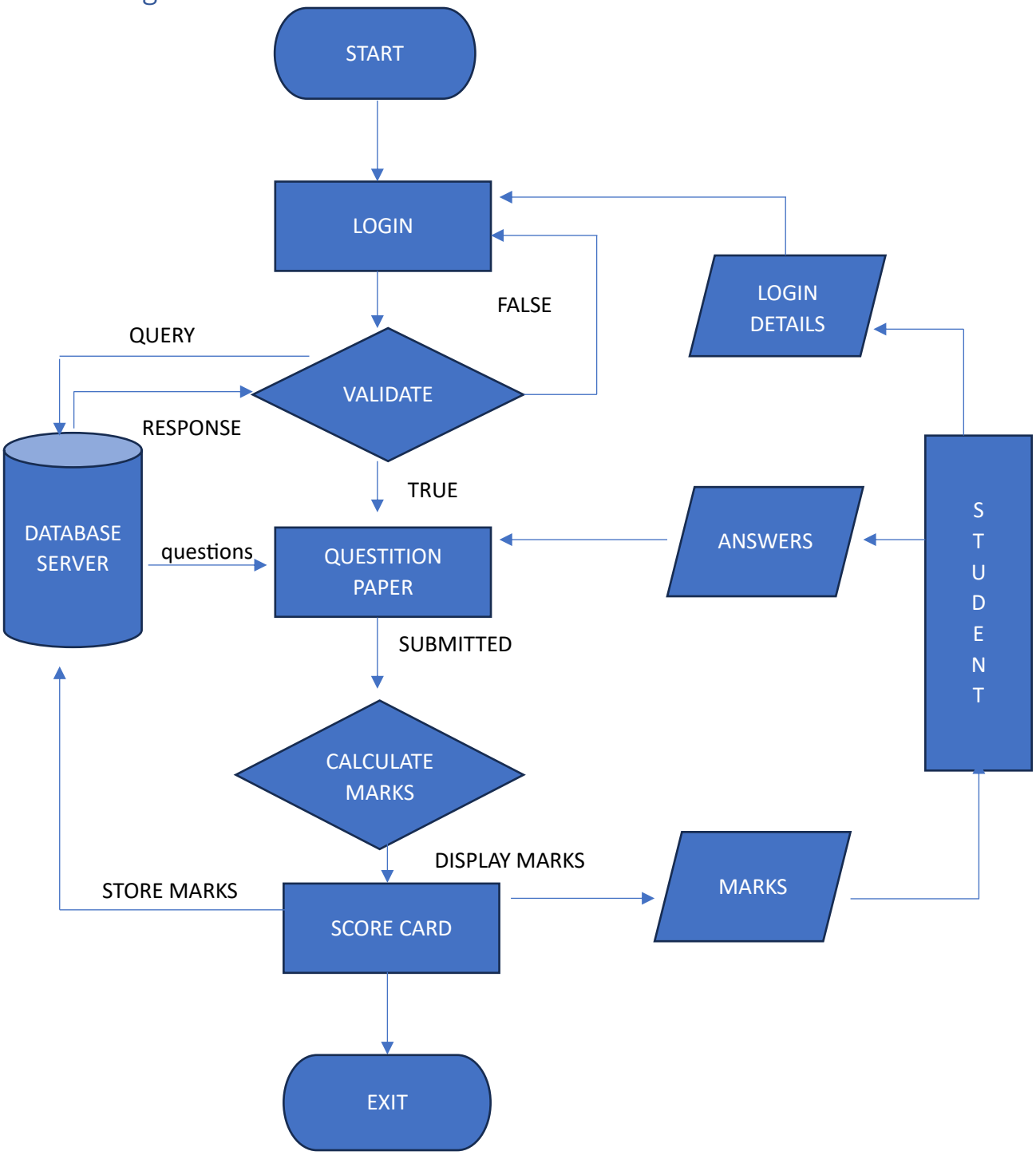
Name: Shetty Tarun Ramesh
Registration Number: 230970005
Programme: MCA
Section : A
Course: Object Oriented Programming -2
FISAC(mini project)

Overview Of the Problem Statement

Education Institution has the following:
A database server with records related to Test takers credentials and a set of records holding Questions.

Requirements:
Query the login collection with the Student input details if successful open question paper.
Retrieval of 5 records of questions from the database server and display to student.
Student user is only permitted to select one option.
Upon submission of the answers evaluate the answers and display the score to student and save the score to database server.

Flow Diagram



Swings Components Used:

1. JFrame - It is a top-level container that represents the main window of an application. It can hold and organize other components such as buttons, text fields, etc. In the project it is used to create three applications login, quiz, and score.
2. JLayeredPane - It is a Swing container that provides a depth-based ordering for its contained components, allowing you to layer components on top of each other. It is used to layer Jlabel which contains an image and another JLayerPanel which holds a set of radio buttons.
3. JPanel - A container that is used to hold and organize other components. It is often used to group related components together. This is used in login page to create the green background.
4. JLabel - A component used to display a single line of read-only text or an image (or both) to the user. It is used to show text to user in many places.
5. JRadioButton - A GUI component that allows the user to choose only one option from a set of mutually exclusive options. It is used to show questions retrieved from the user.
6. ButtonGroup - A logical grouping of radio buttons that ensures that only one radio button in the group can be selected at a time. It is used to group the radio buttons showing questions to the user.
7. JButton - A button that the user can click to perform an action. It is used to submit in login page. Next and submit in questionnaire page.
8. JTextField - A GUI component that allows the user to enter and edit a single line of text. It is used to retrieve username from user.
9. JPasswordField - A text field specialized for password entry, where the characters typed by the user are typically displayed as dots or asterisks to hide them. It is used to retrieve password from user.
10. ImageIcon - An implementation of the Icon interface that paints icons from images. It can be used to display images in various Swing components. It is used to create an Jlabel which holds the image with the doctor.

Events and Actions

1. Constructors – Initialize all components for each frame.
2. nextBtnActionPerformed – When clicked Display follow up questions.
3. submitActionPerformed – When clicked in login validates against database the executes Main of Questionaire. When clicked in Questionaire uploads score and displays scorecard.
4. opaActionPerformed – When the radio button is clicked it enables nextBtn or submitBtn.

Program Code

Index:

1. AuthenticationManager:

```
//Class to connect to MongoDB instance
package com.edu.questionaire;

import com.mongodb.client.MongoClient;
```

```
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.result.UpdateResult;
import org.bson.Document;

public class AuthenticationManager {

    private final MongoCollection<Document> usersCollection;

    // Constructor to initialize MongoDB connection and get the users collection
    public AuthenticationManager() {
        String uri = "mongodb://localhost:27017";
        try {
            // Creating MongoDB client
            MongoClient mongoClient = MongoClients.create(uri);
            // Accessing the MedMCQA database
            MongoDatabase database = mongoClient.getDatabase("MedMCQA");
            // Getting the "Accounts" collection from the database
            usersCollection = database.getCollection("Accounts");
        } catch (Exception e) {
            // If there's an error, throw a runtime exception
            throw new RuntimeException("Failed to initialize MongoDB connection: " +
e.getMessage());
        }
    }

    // Method to update the score of a user in the database
    public boolean UpdateScore(int marks) {
        try {
            // Get username and password from LoginDetails class
            String username = LoginDetails.username, password = LoginDetails.password;

            // Check if username or password is empty
            if (username.isEmpty() || password.isEmpty()) {
                System.out.println("Cannot Update Empty String Error");
                return false;
            } else {
                // Create a filter for username and password
                Document filter = new Document();
                filter.append("username", username);
                filter.append("password", password);

                // Create an update to set the new value of the score field
                Document update = new Document("$set", new Document("score", marks));

                // Update the document that matches the filter
                UpdateResult result = usersCollection.updateOne(filter, update);

                // Check if the update was successful
                return result.getModifiedCount() > 0;
            }
        } catch (NullPointerException e) {
```

```
            System.out.println("String is empty");
            return false;
        }
    }

    // Method to authenticate user based on username and password
    public boolean authenticateUser(String username, String password) {
        // Create a query document to find the user in the collection
        Document query = new Document("username", username)
                .append("password", password);
        // Find the user document based on the query
        Document user = usersCollection.find(query).first();
        // Return true if user exists, false otherwise
        return user != null;
    }

    // Main method for testing authentication
    public static void main(String[] args) {
        // Create an instance of AuthenticationManager
        AuthenticationManager authManager = new AuthenticationManager();
        // Test username and password
        String username = "username@1";
        String password = "password@1";
        // Authenticate the user and print result
        if (authManager.authenticateUser(username, password)) {
            System.out.println("User authentication successful.");
        } else {
            System.out.println("Invalid username or password.");
        }
    }
}
```

2. Login:

```
//Class to display login form

package com.edu.questionaire;
public class Login extends javax.swing.JFrame {

    // Constructor to initialize components and hide the invalid user label
    public Login() {
        initComponents();
        invaliduser.setVisible(false);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
    // Layered Pane for organizing components
    jLayeredPane1 = new javax.swing.JLayeredPane();
    // Password field for entering password
    passText = new javax.swing.JPasswordField();
```

```
    // Layered Pane for styling welcome message
    jLayeredPane2 = new javax.swing.JLayeredPane();
    // Labels for welcome message
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    // Labels for username and password fields
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    // Text field for entering username
    userText = new javax.swing.JTextField();
    // Label for "Login" title
    jLabel6 = new javax.swing.JLabel();
    // Button for submitting credentials
    submit = new javax.swing.JButton();
    // Label for displaying invalid user credentials message
    invaliduser = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setBackground(new java.awt.Color(255, 255, 255));
    // Set icon image for the window
    setIconImage(new Questions().img);

    jLayeredPane1.setBackground(new java.awt.Color(255, 255, 255));
    jLayeredPane1.setOpaque(true);

    // Add password field to the layered pane
    jLayeredPane1.add(passText);
    passText.setBounds(460, 290, 130, 30);

    jLayeredPane2.setBackground(new java.awt.Color(0, 102, 0));
    jLayeredPane2.setOpaque(true);

    // Styling for welcome message
    jLabel3.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
    jLabel3.setForeground(new java.awt.Color(255, 255, 255));
    jLabel3.setText("Welcome To");

    jLabel4.setFont(new java.awt.Font("Segoe UI", 0, 36)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(255, 255, 255));
    jLabel4.setText("MedQuiz");

    jLabel5.setFont(new java.awt.Font("Segoe UI", 0, 24)); // NOI18N
    jLabel5.setForeground(new java.awt.Color(255, 255, 255));
    jLabel5.setText("-a quizing platform");

    // Add welcome message labels to the layered pane
    jLayeredPane2.setLayer(jLabel3, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jLayeredPane2.setLayer(jLabel4, javax.swing.JLayeredPane.DEFAULT_LAYER);
    jLayeredPane2.setLayer(jLabel5, javax.swing.JLayeredPane.DEFAULT_LAYER);

    javax.swing.GroupLayout jLayeredPane2Layout = new
javax.swing.GroupLayout(jLayeredPane2);
```

```
    jLayeredPane2.setLayout(jLayeredPane2Layout);
    jLayeredPane2Layout.setHorizontalGroup(
        jLayeredPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jLayeredPane2Layout.createSequentialGroup()
            .addGap(69, 69, 69)

.addGroup(jLayeredPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILI
NG)
                .addComponent(jLabel4)
                .addComponent(jLabel3))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jLayeredPane2Layout.createSequentialGroup()
            .addContainerGap(48, Short.MAX_VALUE)
            .addComponent(jLabel5,              javax.swing.GroupLayout.PREFERRED_SIZE,              223,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(29, 29, 29))
    );
    jLayeredPane2Layout.setVerticalGroup(
        jLayeredPane2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jLayeredPane2Layout.createSequentialGroup()
            .addGap(123, 123, 123)
            .addComponent(jLabel3,              javax.swing.GroupLayout.PREFERRED_SIZE,              58,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel4,              javax.swing.GroupLayout.PREFERRED_SIZE,              60,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel5,              javax.swing.GroupLayout.PREFERRED_SIZE,              118,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(137, Short.MAX_VALUE))
    );

    // Add styled welcome message to the layered pane
    jLayeredPane1.add(jLayeredPane2);
    jLayeredPane2.setBounds(0, 0, 300, 520);

    // Label for "Password" field
    jLabel1.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
    jLabel1.setText("Password:");
    // Add "Password" label to the layered pane
    jLayeredPane1.add(jLabel1);
    jLabel1.setBounds(360, 290, 90, 30);

    // Label for "Username" field
    jLabel2.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
    jLabel2.setText("Username:");
    // Add "Username" label to the layered pane
    jLayeredPane1.add(jLabel2);
    jLabel2.setBounds(360, 250, 90, 30);

    // Text field for entering username
    userText.setToolTipText("Enter the Username");
```

```
    userText.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            userTextActionPerformed(evt);
        }
    });
    // Add username text field to the layered pane
    jLayeredPane1.add(userText);
    userText.setBounds(460, 250, 130, 30);

    // Label for "Login" title
    jLabel6.setFont(new java.awt.Font("Script MT Bold", 1, 36)); // NOI18N
    jLabel6.setText("Login");
    // Add "Login" label to the layered pane
    jLayeredPane1.add(jLabel6);
    jLabel6.setBounds(420, 150, 100, 70);

    // Button for submitting credentials
    submit.setFont(new java.awt.Font("Segoe UI", 0, 18)); // NOI18N
    submit.setText("Submit");
    submit.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            submitActionPerformed(evt);
        }
    });
    // Add submit button to the layered pane
    jLayeredPane1.add(submit);
    submit.setBounds(410, 360, 110, 40);

    // Label for displaying invalid user credentials message
    invaliduser.setForeground(new java.awt.Color(255, 0, 0));
    invaliduser.setText("Invalid User Credentials");
    // Add invalid user label to the layered pane
    jLayeredPane1.add(invaliduser);
    invaliduser.setBounds(400, 330, 180, 16);

    // Set layout of the window
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLayeredPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 641,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLayeredPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 468,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );

    pack();
} // </editor-fold>
// Submit Button Action Event Handler
    private void submitActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // Create an instance of AuthenticationManager
        AuthenticationManager authManager = new AuthenticationManager();

        // Get username and password from the text fields
        String username = userText.getText();
        String password = passText.getText();

        // Check if username or password is empty
        if (username.isEmpty() || password.isEmpty()) {
            // Show invalid user label and set text
            invaliduser.setVisible(true);
            invaliduser.setText("User cannot be empty");
        } else {
            // Authenticate the user
            if (authManager.authenticateUser(username, password)) {
                // If authentication successful, set username and password in LoginDetails
                LoginDetails.password = password;
                LoginDetails.username = username;
                // Hide login window and show questions window
                setVisible(false);
                Questions.main(new String[]{});
            } else {
                // If authentication failed, show invalid user label and set text
                invaliduser.setVisible(true);
                invaliduser.setText("Invalid username or password.");
                System.out.println("Invalid username or password.");
            }
        }
    }


    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Login().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel invaliduser;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLayeredPane jLayeredPane1;
    private javax.swing.JLayeredPane jLayeredPane2;
    private javax.swing.JPasswordField passText;
    private javax.swing.JButton submit;
    private javax.swing.JTextField userText;
    // End of variables declaration
```

```
}
```

3. LoginDetails:

```
//Class to hold LoginDetails for the Session

package com.edu.questionaire;

public class LoginDetails {
   public static String username;//Static feilds to use as Session
   public static String password;
}
```

4. QuestionManager:

```
//Class to connect to MongoDb Instance and retrieve into Question Object
package com.edu.questionaire;

import static com.mongodb.MongoClientSettings.getDefaultCodecRegistry;

import static org.bson.codecs.configuration.CodecRegistries.fromProviders;
import static org.bson.codecs.configuration.CodecRegistries.fromRegistries;
import org.bson.codecs.configuration.CodecProvider;
import org.bson.codecs.configuration.CodecRegistry;
import org.bson.codecs.pojo.PojoCodecProvider;

import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Aggregates;
import com.mongodb.client.model.Sorts;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class QuestionManager {

   public List<Question> getQuestions()
   {
      List<Question> questionList = new ArrayList<>();

      // Configure POJO codec registry for MongoDB
      CodecProvider pojoCodecProvider = PojoCodecProvider.builder().automatic(true).build();
      CodecRegistry pojoCodecRegistry = fromRegistries(getDefaultCodecRegistry(),
fromProviders(pojoCodecProvider));

      // Replace the placeholder with your MongoDB deployment's connection string
      String uri = "mongodb://localhost:27017";

      try (MongoClient mongoClient = MongoClients.create(uri)) {
         // Connect to the database with the configured codec registry
```

```
        MongoDatabase database = mongoClient.getDatabase("MedMCQA").withCodecRegistry(pojoCodecRegistry);
        // Access the collection of questions
        MongoCollection<Question> collection = database.getCollection("train", Question.class);

        // Sample 5 random documents and sort them by _id
        try (MongoCursor<Question> cursor = collection.aggregate(
            Arrays.asList(Aggregates.sample(5),
                Aggregates.sort(Sorts.ascending("_id"))))
            .iterator()) {
          // Iterate through the cursor and add questions to the list
          while (cursor.hasNext()) {
            questionList.add(cursor.next());
          }
          System.out.println(questionList);
        }
      } catch (Exception e) {
        System.out.println("An error occurred while retrieving questions: " + e.getMessage());
      }

      return questionList;

    }

    public static void main( String[] args ) {
      // Call the getQuestions method when the class is run
      new QuestionManager().getQuestions();
    }
}
```

5. Question:

```
package com.edu.questionaire;

/**
 *Class to hold questions from database
 * @author Tarun
 */
public class Question {

    private String question;
    private String opa;
    private String opb;
    private String opc;
    private int cop;

    // Constructors
    public Question() {
    }

    public Question(String question, String opa, String opb, String opc, int cop) {
        this.question = question;
```

```
        this.opa = opa;
        this.opb = opb;
        this.opc = opc;
        this.cop = cop;
    }

    // Getters and setters
    public String getQuestion() {
        return question;
    }

    public void setQuestion(String question) {
        this.question = question;
    }

    public String getOpa() {
        return opa;
    }

    public void setOpa(String opa) {
        this.opa = opa;
    }

    public String getOpb() {
        return opb;
    }

    public void setOpb(String opb) {
        this.opb = opb;
    }

    public String getOpc() {
        return opc;
    }

    public void setOpc(String opc) {
        this.opc = opc;
    }

    public int getCop() {
        return cop;
    }

    public void setCop(int cop) {
        this.cop = cop;
    }

    @Override
    public String toString() {
        return "Question: " + question + ", Option A: " + opa + ", Option B: " + opb + ", Option C: " +
opc + ", Correct Option: " + cop+"\n";
    }
```

```
}
```

6. Questions

```
//Class extending Jframe to display Questions
package com.edu.questionaire;
import java.awt.Image;
import java.util.List;
import javax.swing.ButtonModel;
import javax.swing.ImageIcon;
/**
 * JFrame for displaying and handling quiz questions.
 * Author: Tarun
 */
public class Questions extends javax.swing.JFrame {
    // Image for the JFrame icon
    Image img = (new
ImageIcon(getClass().getResource("/resources/icons/med_icon225x225.jpeg"))).getImage();
    // List to hold the questions
    List<Question> qlist = new QuestionManager().getQuestions();
    // Index of the current question being displayed
    int currIndex = 0;
    // Array to store user answers
    Integer ans[] = new Integer[5];
    // Array to store correct options
    Integer copt[] = new Integer[5];
    // Static variable to store marks
    private static int marks = 0;
    // Constructor
    public Questions() {
        initComponents();
        showQuestion();
    }
    // Method to display the current question
    private void showQuestion() {

        Question qt = qlist.get(currIndex);
        String qn = " " + (currIndex + 1) + ")" + qt.getQuestion();
        q.setText(qn);
        opa.setText("Option A: " + qt.getOpa());
        opb.setText("Option B: " + qt.getOpb());
        opc.setText("Option C: " + qt.getOpc());
        Qpanel.add(q);
        Qpanel.add(opa);
        Qpanel.add(opb);
        Qpanel.add(opc);
        copt[currIndex] = qt.getCop();
        radiogrp.clearSelection();
        nextBtn.setEnabled(false);
        if (currIndex == 4) {
            nextBtn.setText("Submit");
        }
    }
```

```
// Method to enable the Next button
private void enableNext() {
   nextBtn.setEnabled(true);
}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

   radiogrp = new javax.swing.ButtonGroup();
   jLayeredPane1 = new javax.swing.JLayeredPane();
   Qpanel = new javax.swing.JPanel();
   q = new javax.swing.JLabel();
   opa = new javax.swing.JRadioButton();
   opb = new javax.swing.JRadioButton();
   opc = new javax.swing.JRadioButton();
   nextBtn = new javax.swing.JButton();
   image = new javax.swing.JLabel();

   setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
   setBackground(new java.awt.Color(255, 255, 255));
   setIconImage(img);

   jLayeredPane1.setBackground(new java.awt.Color(255, 255, 255));
   jLayeredPane1.setOpaque(true);

   Qpanel.setBackground(new java.awt.Color(204, 204, 255));
   Qpanel.setToolTipText("");
   Qpanel.setLayout(null);

   q.setBackground(new java.awt.Color(0, 153, 153));
   q.setForeground(new java.awt.Color(255, 255, 255));
   q.setText("Label");
   q.setHorizontalTextPosition(javax.swing.SwingConstants.LEADING);
   q.setMaximumSize(new java.awt.Dimension(100, 42));
   q.setOpaque(true);
   Qpanel.add(q);
   q.setBounds(40, 20, 960, 16);
   q.getAccessibleContext().setAccessibleName("q");

   radiogrp.add(opa);
   opa.setActionCommand("0");
   opa.setLabel("opa");
   opa.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
         opaActionPerformed(evt);
      }
   });
   Qpanel.add(opa);
   opa.setBounds(70, 40, 960, 21);

   radiogrp.add(opb);
   opb.setText("jRadioButton2");
```

```
        opb.setActionCommand("1");
        opb.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                opbActionPerformed(evt);
            }
        });
        Qpanel.add(opb);
        opb.setBounds(70, 70, 960, 21);
        opb.getAccessibleContext().setAccessibleName("opb");

        radiogrp.add(opc);
        opc.setText("jRadioButton3");
        opc.setActionCommand("2");
        opc.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                opcActionPerformed(evt);
            }
        });
        Qpanel.add(opc);
        opc.setBounds(70, 100, 960, 21);
        opc.getAccessibleContext().setAccessibleName("opc");

        nextBtn.setText("Next");
        nextBtn.setEnabled(false);
        nextBtn.setVerticalAlignment(javax.swing.SwingConstants.BOTTOM);
        nextBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                nextBtnActionPerformed(evt);
            }
        });
        Qpanel.add(nextBtn);
        nextBtn.setBounds(40, 140, 72, 23);

        jLayeredPane1.add(Qpanel);
        Qpanel.setBounds(0, 250, 1040, 260);

        image.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resources/icons/medical_bg1200x500.jpg")));  //
NOI18N
        image.setIconTextGap(0);
        jLayeredPane1.add(image);
        image.setBounds(0, 0, 1040, 250);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLayeredPane1,       javax.swing.GroupLayout.PREFERRED_SIZE,       1038,
javax.swing.GroupLayout.PREFERRED_SIZE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(jLayeredPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 509,
Short.MAX_VALUE)
      );

      pack();
   }// </editor-fold>

   private void opaActionPerformed(java.awt.event.ActionEvent evt) {
      enableNext();
   }

   private void opbActionPerformed(java.awt.event.ActionEvent evt) {
      enableNext();
   }

   private void opcActionPerformed(java.awt.event.ActionEvent evt) {
      enableNext();
   }
   // Method to calculate marks
   private void calculateMarks() {
      for (int i = 0; i < 5; i++) {
         if (copt[i].equals(ans[i])) {
            marks++;
         }
      }
   }
   // Getter for marks
   public static int getMarks() {
      return marks;
   }

   // Event handler for Next button
   private void nextBtnActionPerformed(java.awt.event.ActionEvent evt) {
      System.out.println("Index:" + currIndex);
      ButtonModel selectedModel = radiogrp.getSelection();
      Integer selectedText = Integer.valueOf(selectedModel.getActionCommand());
      ans[currIndex] = selectedText;
      if (currIndex == 4) {
         for (Integer s : ans) {
            System.out.println("Ans:" + s);
         }
         for (Integer c : copt) {
            System.out.println("C:" + c);
         }
         calculateMarks();
         System.out.println("Marks:" + marks);
         showGrade();
      }
      if (currIndex < 4) {
         currIndex++;
         showQuestion();
      }
   }
```

```
    // Method to display grade
    private void showGrade() {
        setVisible(false);
        Result.main(new String[]{});
    }

    // Main method to run the application
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            @Override
            public void run() {
                new Questions().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JPanel Qpanel;
    private javax.swing.JLabel image;
    private javax.swing.JLayeredPane jLayeredPane1;
    private javax.swing.JButton nextBtn;
    private javax.swing.JRadioButton opa;
    private javax.swing.JRadioButton opb;
    private javax.swing.JRadioButton opc;
    private javax.swing.JLabel q;
    private javax.swing.ButtonGroup radiogrp;
    // End of variables declaration
}
```

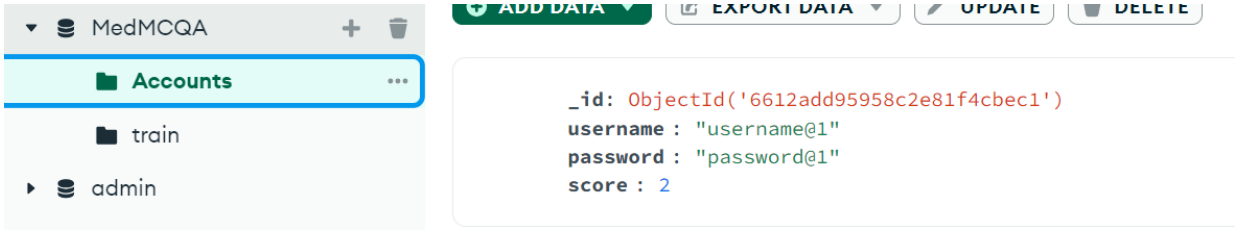
7. Result

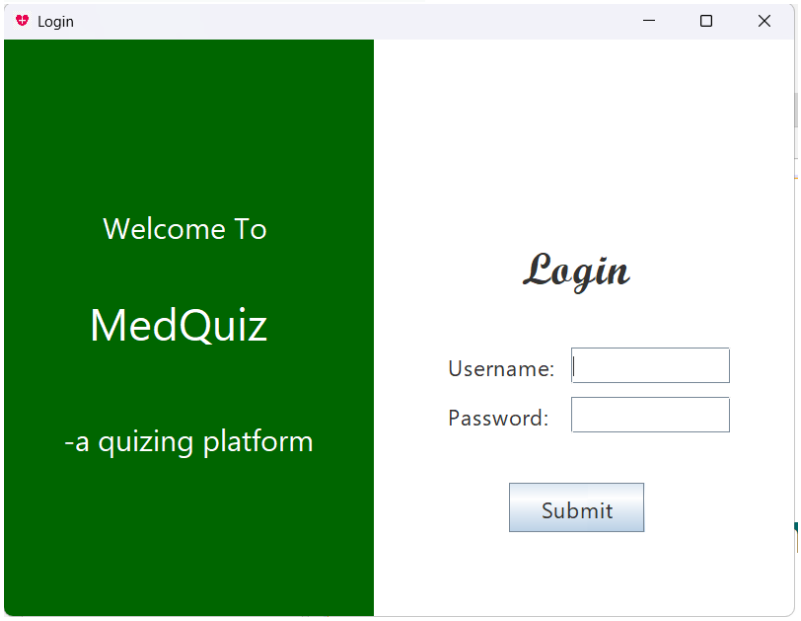
```
//Class to show score and store marks to database
package com.edu.questionaire;

public class Result extends javax.swing.JFrame {

    // Creates new form Result

    // Method to display the score and save it
    private void displayscore() {
        // Set the score label text to display the score
        score.setText("Score:" + Questions.getMarks() + "/5");
        // Print the score to the console
        System.out.println(Questions.getMarks());
        // Call the saveScore method to save the score to the database
        saveScore();
    }

    // Method to save the score to the database
    public void saveScore() {
        // Create an instance of AuthenticationManager to manage authentication and score saving
        AuthenticationManager mgr = new AuthenticationManager();
```

```
        // Update the score in the database and print the result to the console
        if (mgr.UpdateScore(Questions.getMarks())) {
            System.out.println("Successful");
        } else {
            System.out.println("Unsuccessful");
        }
    }

    // Constructor to initialize the Result frame
    public Result() {
        // Initialize the components of the frame
        initComponents();
        // Display the score
        displayscore();
    }

    // Auto-generated code for initializing Swing components
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    // Method to initialize Swing components
private void initComponents() {

    // Create a layered pane to hold components
    jLayeredPane1 = new javax.swing.JLayeredPane();
    // Label to display submission message
    jLabel1 = new javax.swing.JLabel();
    // Label to display tick mark icon
    jLabel2 = new javax.swing.JLabel();
    // Label to display the score
    score = new javax.swing.JLabel();

    // Set the default close operation and icon image for the frame
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setIconImage(new Questions().img);

    // Set background color for the layered pane
    jLayeredPane1.setBackground(new java.awt.Color(255, 255, 255));
    jLayeredPane1.setOpaque(true);

    // Set properties for the submission message label
    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 48));
    jLabel1.setForeground(new java.awt.Color(0, 102, 102));
    jLabel1.setText("Your Answers have been Submitted!");
    jLayeredPane1.add(jLabel1);
    jLabel1.setBounds(40, 210, 850, 146);

    // Set properties for the tick mark icon label
    jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/com/edu/resources/tick_mark.jpg")));
    jLayeredPane1.add(jLabel2);
    jLabel2.setBounds(10, 10, 360, 260);

    // Set properties for the score label
```

```
        score.setFont(new java.awt.Font("Segoe UI", 0, 36));
        score.setText("Score");
        jLayeredPane1.add(score);
        score.setBounds(440, 100, 320, 70);

        // Set layout for the frame
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLayeredPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 922,
Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLayeredPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 443,
Short.MAX_VALUE)
        );

        // Auto-size the frame
        pack();
    }// </editor-fold>
        // Variables declaration - do not modify
        private javax.swing.JLabel jLabel1;
        private javax.swing.JLabel jLabel2;
        private javax.swing.JLayeredPane jLayeredPane1;
        private javax.swing.JLabel score;
        // End of variables declaration
    }
```

Screenshots

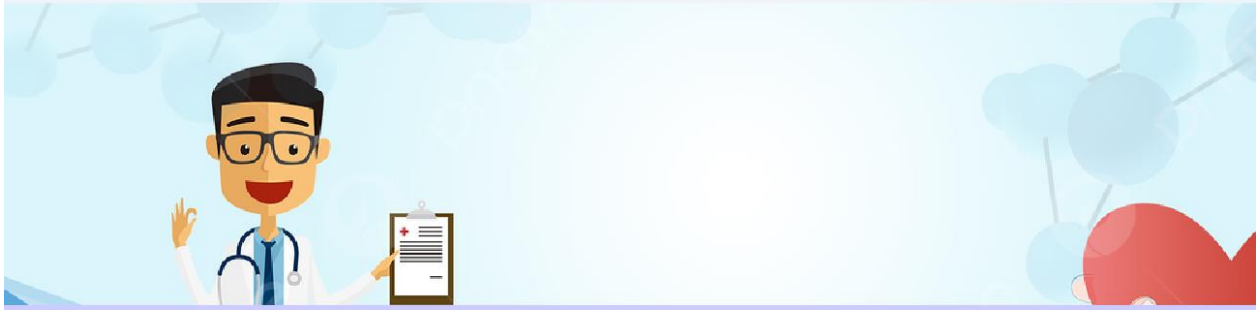

MedMCQA
Accounts
train
admin
ADD DATA
EXPORT DATA
UPDATE
DELETE
_id: ObjectId('6612add95958c2e81f4cbec1')
username : "username@1"
password : "password@1"
score : 2


Login
Welcome To
MedQuiz
-a quizing platform
Login
Username:
Password:
Submit


Login
Username:
Password:
Enter the Usernar
User cannot be empty
Submit


Login
Username:
adfda
Password:
*****
User cannot be empty
Submit

Questions



1)Growth hormone has its effect on growth through?

- Option A: Directly
- Option B: IG1-1
- Option C: Thyroxine

Next

Questions

5)IgA deposits on skin biopsy

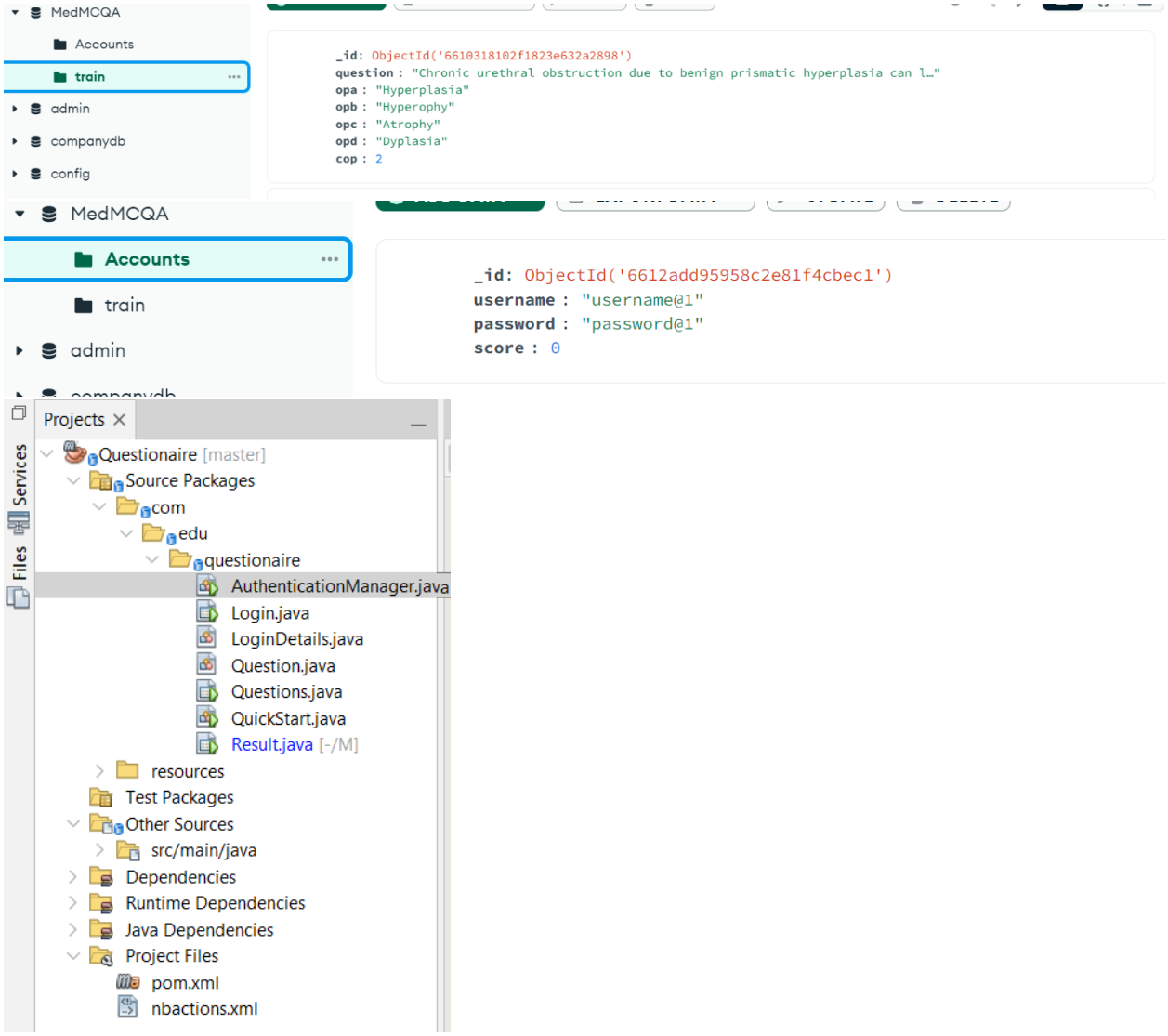
- Option A: Henoch Schouleiln puspura
- Option B: Giant cell aeritis
- Option C: Microscopic polyangitis

Submit

Result

Score:2/5

Your Answers have been Submitted!



References

1) Oracle - https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/overview-summary.html
2) Java™ : The Complete Reference, Seventh Edition
3) Tutorials Point
4) Stack Overflow
5) NetBeans - https://netbeans.apache.org/
6) Mongodb.com
7) Youtube.com