SGCP's
Guru Nanak Khalsa College
of Arts, Science & Commerce (Autonomous)

Mini-Project:
Protein Research Toolkit

Roll No.	Team Member
110	Kasturi Nayan Prabhakar Anuradha
115	Kothari Prarthi Hrishit Dhara
119	Pal Khushi Ramsharan Mithlesh

Course: M. Sc. Bioinformatics (Part I)

Department: Department of Bioinformatics

Academic Year: 2023-24

Paper Code: GNKPSBI2P502 (CHECK)

Paper Name: Mandatory Paper II: Java Programming, Introduction to Linux and Machine Learning

AIM OF THE PROJECT:

The primary objective of this project is to develop a comprehensive and user-friendly interface platform known as the Protein Research Toolkit. Built using Java, Swing, and JDBC, this toolkit aims to empower researchers with a powerful set of tools for in-depth protein analysis. By providing a smooth and intuitive user experience, the Protein Research Toolkit strives to streamline researchers' workflows and enhance their efficiency in protein-centric inquiries. In addition to offering a wide range of features, this platform will prioritize ease-of-use, ensuring that researchers of all experience levels can leverage its capabilities for their protein investigations.

COMPONENTS OF THE PROJECT AND THEIR FUNCTIONALITIES:

The Protein Research Toolkit project consists of several key components designed to provide researchers with a comprehensive suite of functionalities for protein analysis. These components include:

1. **User Login and Credential Localization:**
   The Protein Research Toolkit implements a secure user registration and login system utilizing email addresses and passwords to manage user accounts and preferences along with ensuring data privacy. Upon registration, a unique license key will be automatically generated for each user. All user credentials and protein analysis data will be securely stored within a dedicated SQL database as well as at a secured local location, potentially allowing for personalized workflows in the future.

2. **Protein Sequence Search:**
   Upon inputting and searching for a protein query of interest in the designated textfield, the protein sequence search component returns detailed results for each identified protein fetched from the UniProt database, showcasing:
   a. **UniProt ID:** Unique identifier in the Universal Protein Resource database.
   b. **Protein Entry Name:** Descriptive name assigned to the protein.
   c. **Sequence Length:** Length of the protein sequence in amino acids (in bp).
   d. **Organism:** Source organism from which the protein originates.
   e. **Gene Name:** Name of the gene encoding the protein.
   f. **FASTA Sequence:** Full amino acid sequence of the protein in FASTA format.

3. **Structure Visualization:**
   The structure visualization module will display a list of retrieved proteins from the Protein Data Bank (PDB) database, including the PDB ID, protein entry name, and the method used for structure determination. Users can then select a protein from the list to view its 3D structure in an interactive viewer, JMol. Additionally, the module provides the option to directly send the PDB IDs of selected proteins from the list to the next tab (Structural Alignment) for further analysis.

4. **Structural Alignment:**
   Through the comprehensive Structural Alignment module, Users can select and input PDB IDs of proteins of interest from the previous tab, 'Structure Visualization', to perform a multiple sequence alignment using FATCAT (flexible) algorithm. Results are presented in two ways:

   a. **Aligned Sequences:** The aligned sequences are displayed in a dedicated text field, allowing for easy comparison of amino acid residues across the proteins.
   b. **Interactive 3D Visualization:** The aligned portions of the protein structures are visualized in 3D. This visualization provides users with in-depth insights into structural similarities and differences. Users can further customize the view by:
      - **Style:** Changing the representation of protein structures (e.g., cartoons).
      - **Color:** Selecting different color schemes to highlight specific features (e.g., by amino acid type, hydrophobicity).
      - **Palette:** Modifying the color palette for a tailored visualization experience.

   Additionally, hovering the mouse over a specific residue in the 3D view will display the corresponding amino acid type.
   For broader evolutionary context, the 'View' tab offers the option to generate and visualize a phylogenetic tree based on the aligned sequences. This functionality allows users to explore the evolutionary relationships between the analyzed proteins.

5. **Literature Search:**
   Using the literature search functionality, the search results for a particular term displays a comprehensive list of relevant research papers, including:
   a. **DOI ID:** A unique identifier for easy retrieval of the full article.
   b. **Title:** The title of the research paper for quick identification of relevant content.
   c. **Authors:** List of authors who conducted the research.
   d. **Year of Publication:** The year the research was published to help users gauge the recency of the information.
   e. **Citation Count:** The number of times the research paper has been cited by other studies, indicating its potential impact in the field.
   f. **Link to the Article:** A direct link to the full text of the research paper, allowing users to seamlessly access the complete source material.

6. **Citation Generator:**
   Users can input a DOI ID for both research papers and books. This versatile tool will then automatically generate formatted citations for the chosen reference in three widely-used styles: APA (7^{th} ed.), MLA (9^{th} ed.), and Harvard (International). This eliminates the need for manual formatting and ensures bibliographic accuracy.

FUTURE PROSPECTS:

By integrating with existing bioinformatics tools, the Protein Research Toolkit can provide a comprehensive platform for protein analysis and research, enabling users to access a wide range of tools and resources in a single platform.

1. **Integration with sequence databases:** The Protein Research Toolkit can be integrated with sequence databases such as the non-redundant protein or nucleotide databases at NCBI, by providing tools for identifier-based sequence retrieval.
2. **Integration with public bioinformatics tools:** The Protein Research Toolkit can be integrated with existing public bioinformatics tools ClustalW, and MUSCLE that would enable users to access these tools without the need to install them separately.
3. **Integration with in-house bioinformatics tools:** The Protein Research Toolkit can also be integrated with in-house bioinformatics tools developed by research groups.
4. **Integration with protein structure prediction programs:** The Protein Research Toolkit can be integrated with protein structure visualization programs such as AlphaFold2, which can be used to predict the protein structures based on their sequences.
5. **Integration with web-based bioinformatics tools:** The Protein Research Toolkit can be integrated with web-based bioinformatics tools such as the BioJS MSA viewer which can be used to display multiple sequence alignments.

CODE:

Package	Files
Backend	ProcessDialog.java
	SQLTest.java
	UniProtSearch.java
	PDBSearch.java
	VisStruct.java
	StructAli.java
	LiteratureSearch.java
	GenerateCitation.java
Frontend	MainFrame.java

ProcessDialog.java

```
package Backend;

import Frontend.MainFrame;
import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.BorderFactory;
import javax.swing.JDialog;
import javax.swing.JLabel;
```

```
import javax.swing.JPanel;
import javax.swing.JProgressBar;
import javax.swing.SwingConstants;
import javax.swing.SwingUtilities;

public class ProcessDialog {
   private MainFrame mainFrame;

   public ProcessDialog(MainFrame mainFrame) {
      this.mainFrame = mainFrame;
   }

   public void processDialog(Runnable function) {
      JDialog processingDialog = new JDialog();
      JLabel processingLabel = new JLabel("Processing...\nPlease Wait",
SwingConstants.CENTER);
      JProgressBar progressBar = new JProgressBar();
      progressBar.setPreferredSize(new Dimension(100, 10));
      progressBar.setIndeterminate(true);
      JPanel panel = new JPanel(new BorderLayout());
      panel.setBorder(BorderFactory.createEmptyBorder(10, 10, 50, 10));
      panel.add(processingLabel, BorderLayout.CENTER);
      panel.add(progressBar, BorderLayout.SOUTH);
      processingDialog.setContentPane(panel);
      processingDialog.setSize(300, 200);
      processingDialog.setLocationRelativeTo(null);
      processingDialog.setModal(true);
      processingDialog.setTitle("Processing");
      processingDialog.setAlwaysOnTop(true);
      Thread processingThread = new Thread(() -> {
         function.run();
         SwingUtilities.invokeLater(() -> processingDialog.dispose());
      });
      processingThread.start();
      processingDialog.setVisible(true);
   }
}
```

SQLTest.java

```
package Backend;

import Frontend.MainFrame;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.Scanner;


public class SQLTest {
    private MainFrame mainFrame;

    public SQLTest(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }

    Scanner sc = new Scanner(System.in);
    String sql_email_id, sql_password, sql_license_key;
    int choice;

    public String generateLicenseKey() {
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMMddHHmmss");
        String formattedDate = dateFormat.format(new java.util.Date());
        return "PTK" + formattedDate + "NPK";
    }

    public void savedata(String email, String pass, String key){
        String[] words = {email, pass, key};
        String filePath = "./assets/Credentials.txt";
        try {
            try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {
            } catch (IOException e) {
                e.printStackTrace();
            }
            BufferedWriter writer = new BufferedWriter(new FileWriter(filePath));
            for (String word : words) {
                writer.write(word);
                writer.newLine();
            }
            writer.close();
```

```
            System.out.println("Words have been written to the file successfully.");
        } catch (IOException e) {
            System.err.println("Error writing to the file: " + e.getMessage());
        }
    }

    public void login()
    {
        try {
            boolean user_input = true;
            String email_id = mainFrame.jTextField5.getText();
            String password = mainFrame.jTextField6.getText();
            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("Registered");
            Connection con;
            Statement smt;
            con =
DriverManager.getConnection("jdbc:mysql://tgj.h.filess.io:3307/ProteinToolkit_positivewe",
"ProteinToolkit_positivewe", "7cfab746c1ffd2b58544cf136cb5fbdc192d9f7b");
            System.out.println("Connection Successful");
            smt = con.createStatement();
            String sql = "SELECT * FROM APPDATA WHERE EMAIL_ID = '" + email_id +
"';";
            ResultSet rs = smt.executeQuery(sql);
            while(user_input)
            {
                if (rs.next()) {
                    sql_email_id = rs.getString("EMAIL_ID");
                    sql_password = rs.getString("PASSWORD");
                    sql_license_key = rs.getString("LICENSE_KEY");
                    if (email_id.equals(sql_email_id) && password.equals(sql_password)) {
                        System.out.println("Login Successful.");
                        mainFrame.jLabel9.setText("Login Succesful");
                        user_input = false;
                    }
                    else {
                        mainFrame.jLabel9.setText("Credentials do not match with Email ID");
                        System.out.println("Credentials do not match with email id.");
                    }
                }
                else {
                    mainFrame.jLabel9.setText("User not found. Please Register");
                    System.out.println("User not found. Register to continue.");
                    user_input = false;
```

```
            }
        }
        savedata(sql_email_id, sql_password, sql_license_key);
        mainFrame.jLabel2.setText(sql_email_id);
        mainFrame.jLabel3.setText(sql_license_key);
        mainFrame.jLabel4.setText("Registered");
        rs.close();
        smt.close();
        con.close();
      }
      catch(SQLException se)
      {
        se.printStackTrace();
      }
      catch(Exception e)
      {
        e.printStackTrace();
      }
    }

    public void register()
    {
      try
      {
        String email_id = mainFrame.jTextField5.getText();
        String password = mainFrame.jTextField6.getText();
        String rlicense_key = generateLicenseKey();
        Class.forName("com.mysql.cj.jdbc.Driver");
        System.out.println("Registered");
        Connection con;
        Statement smt;
        con =
DriverManager.getConnection("jdbc:mysql://tgj.h.filess.io:3307/ProteinToolkit_positivewe",
"ProteinToolkit_positivewe", "7cfab746c1ffd2b58544cf136cb5fbdc192d9f7b");
        System.out.println("Connection Successful");
        smt = con.createStatement();
        System.out.println("Registration Successful.");
        String sql1 = "INSERT INTO APPDATA VALUES('" + email_id + "','" + password +
"','" + rlicense_key + "');";
        smt.executeUpdate(sql1);
        String sql2 = "SELECT * FROM APPDATA WHERE EMAIL_ID = '" + email_id +
"';";
        ResultSet rs = smt.executeQuery(sql2);
        if (rs.next()) {
```

```
            String sql_remail_id = rs.getString("EMAIL_ID");
            String sql_rpassword = rs.getString("PASSWORD");
            String sql_rlicense_key = rs.getString("LICENSE_KEY");
            System.out.println("*FINAL DETAILS POST REGISTRATION*");
            System.out.println("Email ID: " + sql_remail_id);
            System.out.println("Password: " + sql_rpassword);
            System.out.println("License Key: " + sql_rlicense_key);
            savedata(sql_remail_id, sql_rpassword, sql_rlicense_key);
          }
          mainFrame.jTextField7.setText(rlicense_key);
          mainFrame.jLabel9.setText("Registration Succesfull");
          smt.close();
          con.close();
        }

        catch(SQLException se)
        {
          se.printStackTrace();
        }
        catch(Exception e)
        {
          e.printStackTrace();
        }
    }
}
```

UniProtSearch.java

```
package Backend;

import Frontend.MainFrame;
import javax.swing.table.DefaultTableModel;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.List;

public class UniProtSearch {
    private MainFrame mainFrame;
```

```
    public UniProtSearch(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }

    private List<String[]> fetchData(String urlString) throws IOException {
        URL url = new URL(urlString);
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setRequestMethod("GET");
        BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
        String line;
        List<String[]> data = new ArrayList<>();
        while ((line = reader.readLine()) != null) {
            String[] columns = line.split("\t");
            data.add(columns);
        }
        reader.close();
        connection.disconnect();
        return data;
    }

    public void dispUniProt() throws UnsupportedEncodingException {
        String pquery = URLEncoder.encode(mainFrame.getUniSearchText(), "UTF-8");
        String url =
"https://rest.uniprot.org/uniprotkb/search?download=true&fields=accession%2Cid%2Cgene_
names%2Corganism_name%2Clength%2Csequence&format=tsv&query=%28" + pquery +
"%29&size=50";
        DefaultTableModel model = mainFrame.getTableModel();
        model.setRowCount(0);
        try {
            List<String[]> data = fetchData(url);
            if (data != null && !data.isEmpty()) {
                for (int i = 1; i < data.size(); i++) {
                    String[] row = data.get(i);
                    String entry = row[0];
                    String entryName = row[1];
                    String geneNames = row[2];
                    String organism = row[3];
                    String length = row[4];
                    String seq = row[5];
                    model.addRow(new Object[]{entry, entryName, length, organism, geneNames,
seq});
                }
```

```
            } else {
                System.out.println("No data found.");
            }
        } catch (IOException e) {
            model.addRow(new Object[]{"No data found.", "No data found.", "No data found.",
"No data found.", "No data found.", "No data found."});
        }
    }
}
```

PDBSearch.java

```
package Backend;

import Frontend.MainFrame;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.List;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
import org.json.JSONArray;
import org.json.JSONObject;

public class PDBSearch {
    private MainFrame mainFrame;

    public PDBSearch(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }

    public String[] getPDBID() throws UnsupportedEncodingException {
        List<String> identifiersList = new ArrayList<>();
        String query = URLEncoder.encode(mainFrame.getPDBSearchText(), "UTF-8");
        DefaultTableModel model = (DefaultTableModel) mainFrame.jTable2.getModel();
        try {
            String urlEndpoint =
"https://search.rcsb.org/rcsbsearch/v2/query?json=%7B%22query%22%3A%7B%22type%2
2%3A%22terminal%22%2C%22service%22%3A%22full_text%22%2C%22parameters%22
```

```
%3A%7B%22value%22%3A%22"+query+"%22%7D%7D%2C%22request_options%22%3
A%7B%22paginate%22%3A%7B%22start%22%3A0%2C%22rows%22%3A50%7D%7D%
2C%22return_type%22%3A%22entry%22%7D";
            URL url = new URL(urlEndpoint);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            StringBuilder response = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {
               response.append(line);
            }
            reader.close();
            JSONObject jsonResponse = new JSONObject(response.toString());
            JSONArray resultSet = jsonResponse.getJSONArray("result_set");
            for (int i = 0; i < resultSet.length(); i++) {
               JSONObject entry = resultSet.getJSONObject(i);
               String identifier = entry.getString("identifier");
               identifiersList.add(identifier);
            }
            conn.disconnect();
         } catch (Exception e) {
            e.printStackTrace();
            model.addRow(new Object[]{"No data found.", "No data found.", "No data found.",
"No data found.", "No data found."});
         }
         String[] identifiersArray = identifiersList.toArray(new String[identifiersList.size()]);
         return identifiersArray;
      }


      public void dispPDB(String PDBID) throws IOException {
         String endpoint = "https://data.rcsb.org/rest/v1/core/entry/"+PDBID;
         DefaultTableModel model = (DefaultTableModel) mainFrame.jTable2.getModel();
         URL url = new URL(endpoint);
         HttpURLConnection connection = (HttpURLConnection) url.openConnection();
         connection.setRequestMethod("GET");
         BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
         StringBuilder response = new StringBuilder();
         String line;
         while ((line = reader.readLine()) != null) {
            response.append(line);
```

```
        }
        reader.close();
        JSONObject jsonResponse = new JSONObject(response.toString());
        String rcsbId = jsonResponse.getString("rcsb_id");
        String title = jsonResponse.getJSONObject("struct").getString("title");
        JSONArray exptlArray = jsonResponse.getJSONArray("exptl");
        StringBuilder methods = new StringBuilder();
        for (int i = 0; i < exptlArray.length(); i++) {
            JSONObject exptlObj = exptlArray.getJSONObject(i);
            String method = exptlObj.getString("method");
            methods.append(method).append(", ");
        }
        if (methods.length() > 0) {
            methods.setLength(methods.length() - 2);
        }
        SwingUtilities.invokeLater(() -> {
            model.addRow(new Object[]{rcsbId, title, methods.toString(), "Visualize in 3D",
"Send to Structure Alignment"});
        });
        connection.disconnect();
    }
}
```

VisStruct.java

```
package Backend;

import Frontend.MainFrame;
import org.biojava.nbio.structure.Structure;
import org.biojava.nbio.structure.gui.BiojavaJmol;
import org.biojava.nbio.structure.io.PDBFileReader;

public class VisStruct {
    private MainFrame mainFrame;

    public VisStruct(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }

    public void VisStruct(String PDBID) {
        try {
            PDBFileReader pdbr = new PDBFileReader();
            System.setProperty("PDB_DIR", "./assets");
            String pdbCode = PDBID;
```

```
            Structure struc = pdbr.getStructureById(pdbCode);
            BiojavaJmol jmolPanel = new BiojavaJmol();
            jmolPanel.setStructure(struc);
            jmolPanel.evalString("select * ; backbone on;");
            jmolPanel.evalString("select * ; color chain;");
            jmolPanel.evalString("select *; spacefill off; wireframe off; backbone 0.4;  ");
            jmolPanel.evalString("save STATE state_1");
            jmolPanel.evalString("spin on;");
        } catch (Exception e){
            e.printStackTrace();
            }
    }
}
```

StructAli.java

```
package Backend;

import Frontend.MainFrame;
import java.util.*;
import org.biojava.nbio.structure.*;
import org.biojava.nbio.structure.align.StructureAlignment;
import org.biojava.nbio.structure.align.StructureAlignmentFactory;
import org.biojava.nbio.structure.align.fatcat.FatCatFlexible;
import org.biojava.nbio.structure.align.gui.MultipleAlignmentJmolDisplay;
import org.biojava.nbio.structure.align.multiple.MultipleAlignment;
import org.biojava.nbio.structure.align.multiple.mc.MultipleMcMain;
import org.biojava.nbio.structure.align.multiple.util.MultipleAlignmentWriter;
import org.biojava.nbio.structure.align.util.AtomCache;

public class StructAli {
    private MainFrame mainFrame;
    public StructAli(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }

    public void StructAli(String PDBIDs) {
        System.setProperty("PDB_DIR", "./assets");
        List<String> names = new ArrayList<>();
        for (String name : mainFrame.jTextArea1.getText().split("\n")) {
            names.add(name);
        }
        try {
            AtomCache cache = new AtomCache();
```

```
            List<Atom[]> atomArrays = new ArrayList<Atom[]>();
            List<StructureIdentifier> identifiers = new ArrayList<StructureIdentifier>();
            for (String name:names)        {
              atomArrays.add(cache.getAtoms(name));
              identifiers.add(new SubstructureIdentifier(name));
            }
            StructureAlignment pairwise  =
StructureAlignmentFactory.getAlgorithm(FatCatFlexible.algorithmName);
            MultipleMcMain multiple = new MultipleMcMain(pairwise);
            MultipleAlignment result = multiple.align(atomArrays);
            result.getEnsemble().setStructureIdentifiers(identifiers);
            System.out.println(MultipleAlignmentWriter.toFASTA(result));
            mainFrame.jTextArea2.setText(MultipleAlignmentWriter.toFASTA(result));
            MultipleAlignmentJmolDisplay.display(result);
        } catch (Exception e) {
            e.printStackTrace();
            mainFrame.jTextArea2.setText("No Structural Alignments found between "+ names);
        }
    }
}
```

LiteratureSearch.java

```
package Backend;

import Frontend.MainFrame;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
import org.json.JSONArray;
import org.json.JSONObject;

public class LiteratureSearch {
    private MainFrame mainFrame;

    public LiteratureSearch(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }
```

```
    public static boolean isPDF(String urlString) {
        try {
            URL url = new URL(urlString);
            URLConnection connection = url.openConnection();

            BufferedReader reader = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
            String line;
            StringBuilder pageSource = new StringBuilder();
            while ((line = reader.readLine()) != null) {
                pageSource.append(line).append("\n");
            }
            reader.close();
            String pageSourceString = pageSource.toString();
            if (pageSourceString != null && (pageSourceString.contains("application/pdf") ||
pageSourceString.contains("application/x-pdf"))) {
                return true;
            }
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
        return false;
    }

    public void ArticleSearch(String query) {
        String actualquery = query.replace(" ", "+");
        try {
            String apiUrl =
"https://api.crossref.org/works?query.title="+actualquery+"&select=DOI,title,author,issued,is
-referenced-by-count&rows=100&mailto=support@crossref.org";
            URL url = new URL(apiUrl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            conn.setRequestProperty("Accept", "application/json");
            if (conn.getResponseCode() != 200) {
                throw new RuntimeException("Failed : HTTP error code : " +
conn.getResponseCode());
            }
            BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));
            StringBuilder output = new StringBuilder();
            String line;
            while ((line = br.readLine()) != null) {
```

```
            output.append(line);
        }
        conn.disconnect();
        JSONObject jsonObject = new JSONObject(output.toString());
        JSONArray items = jsonObject.getJSONObject("message").getJSONArray("items");
        DefaultTableModel model = (DefaultTableModel) mainFrame.jTable3.getModel();
        model.setRowCount(0);
        for (int i = 0; i < items.length(); i++) {
            JSONObject item = items.getJSONObject(i);
            String doi = item.optString("DOI", "Data not found");
            String title = item.getJSONArray("title").optString(0, "Data not found");
            JSONArray authorsArray = item.optJSONArray("author");
            StringBuilder authors = new StringBuilder();
            if (authorsArray != null) {
                for (int j = 0; j < authorsArray.length(); j++) {
                    JSONObject author = authorsArray.getJSONObject(j);
                    if (j != 0) {
                        authors.append(", ");
                    }
                    authors.append(author.getString("given")).append("
").append(author.getString("family"));
                }
            } else {
                authors.append("Data not found");
            }
            String issuedDate = item.optJSONObject("issued")
                    .optJSONArray("date-parts")
                    .optJSONArray(0)
                    .join(", ");
            issuedDate = issuedDate.isEmpty() ? "Data not found" : issuedDate;
            final String date = issuedDate;
            String timescited = item.optString("is-referenced-by-count", "Data not found");
            String articleURL = "https://sci-hub.se/" + doi;
            if (isPDF(articleURL)==true) {
                articleURL = "https://sci-hub.se/" + doi;
            } else {
                articleURL = "https://dx.doi.org/" + doi;
            }
            final String aURL = articleURL;
            SwingUtilities.invokeLater(() -> {
                model.addRow(new Object[]{doi, title,authors.toString(), date, timescited,
aURL});
            });
        }
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

GenerateCitation.java

```
package Backend;

import Frontend.MainFrame;
import org.json.JSONArray;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;

public class GenerateCitation {

    private MainFrame mainFrame;

    public GenerateCitation(MainFrame mainFrame) {
        this.mainFrame = mainFrame;
    }

    private static String getmlaMonthName(int month) {
        String[] monthNames = {"Jan.", "Feb.", "Mar.", "Apr.", "May", "June", "July", "Aug.",
"Sept.", "Oct.", "Nov.", "Dec."};
        return monthNames[month - 1];
    }

    public void generateArticleCitation(String query) {
        String encodedquery = null;
        try {
            encodedquery = URLEncoder.encode(query, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
```

```
        String citationURL =
"https://www.mybib.com/api/autocite/search?q="+encodedquery+"&sourceId=article_journal
";
        try {
            URL url = new URL(citationURL);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            conn.setRequestProperty("Accept", "application/json");
            BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));
            StringBuilder responseBuilder = new StringBuilder();
            String output;
            while ((output = br.readLine()) != null) {
               responseBuilder.append(output);
            }
            JSONObject jsonResponse = new JSONObject(responseBuilder.toString());
            JSONArray resultsArray = jsonResponse.getJSONArray("results");
            JSONObject result = resultsArray.getJSONObject(0);
            JSONObject metadata = result.getJSONObject("metadata");
            JSONArray authors = metadata.optJSONArray("author");
            StringBuilder authorNames = new StringBuilder();
            if (authors != null) {
               for (int i = 0; i < authors.length(); i++) {
                  JSONObject author = authors.getJSONObject(i);
                  String givenName = author.optString("given", "");
                  String familyName = author.optString("family", "");
                  if (!givenName.isEmpty() || !familyName.isEmpty()) {
                     if (i > 0) {
                        authorNames.append(", ");
                     }
                     authorNames.append(familyName).append(", ").append(givenName);
                  }
               }
            }
            String title = metadata.optString("title", "");
            String containerTitle = metadata.optString("containerTitle", "");
            String volume = metadata.optString("volume", "");
            String issue = metadata.optString("issue", "");
            String page = metadata.optString("page", "");
            String doiValue = metadata.optString("doi", "");
            JSONObject issuedDate = metadata.optJSONObject("issued");
            int year = issuedDate != null ? issuedDate.optInt("year", 0) : 0;
```

```
        String aapacitation = authorNames.toString() + " (" + year + "). " + title + ". " +
containerTitle + ", " + volume + "(" + issue + "), " + page + ". " + "https://doi.org/" +
doiValue;
        String amlacitation = authorNames.toString() + " \"" + title + ".\" " + containerTitle +
", vol. " + volume + ", no. " + issue + ", " + getmlaMonthName(issuedDate.getInt("month"))
+ " " + year + ", pp. " + page + ", " + "https://doi.org/" + doiValue + ".";
        String aharwardcitation = authorNames.toString() + " (" + year + "). " + title + ". " +
containerTitle + ", " + volume + "(" + issue + "), pp." + page + ". doi:" + "https://doi.org/" +
doiValue + ".";
            mainFrame.jTextArea3.setText(aapacitation);
            mainFrame.jTextArea4.setText(amlacitation);
            mainFrame.jTextArea5.setText(aharwardcitation);
            conn.disconnect();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

MainFrame.java

```
package Frontend;

import Backend.*;

import com.formdev.flatlaf.themes.FlatMacLightLaf;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.table.DefaultTableModel;

public class MainFrame extends javax.swing.JFrame {
    private UniProtSearch uni;
    private PDBSearch pdb;
    private VisStruct vs;
    private StructAli sa;
    private ProcessDialog pd;
```

```
private LiteratureSearch as;
private GenerateCitation gc;
private SQLTest st;

public MainFrame() {
    initComponents();
    uni = new UniProtSearch(this);
    pdb = new PDBSearch(this);
    vs = new VisStruct(this);
    sa = new StructAli(this);
    pd = new ProcessDialog(this);
    as = new LiteratureSearch(this);
    gc = new GenerateCitation(this);
    st = new SQLTest(this);
}
public String getUniSearchText(){
    return jTextField1.getText();
}
public String getPDBSearchText(){
    return jTextField2.getText();
}
public String getPMCSearchText(){
    return jTextField3.getText();
}
public DefaultTableModel getTableModel() {
    return (DefaultTableModel) jTable1.getModel();
}
public void prntStructData() throws UnsupportedEncodingException{
    ((DefaultTableModel) jTable2.getModel()).setRowCount(0);
    String[] pdbid = pdb.getPDBID();
    for (int i = 0; i < pdbid.length; i++){
        try {
            String spdbid = pdbid[i];
            pdb.dispPDB(spdbid);
        } catch (IOException ex) {
            Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
public void verify() {
    String filePath = "./assets/Credentials.txt";
    List<String> lines = new ArrayList<>();
    try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
        String line;
```

```
        while ((line = br.readLine()) != null) {
            lines.add(line);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    if (lines.size() >= 2) {
        jLabel2.setText(lines.get(0));
        jLabel3.setText(lines.get(1));
        jLabel4.setText("Registered");
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel15 = new javax.swing.JLabel();
    jButton6 = new javax.swing.JButton();
    jTabbedPane1 = new javax.swing.JTabbedPane();
    UniProtSearch = new javax.swing.JPanel();
    jTextField1 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    PDBSearch = new javax.swing.JPanel();
    jTextField2 = new javax.swing.JTextField();
    jButton5 = new javax.swing.JButton();
    jScrollPane7 = new javax.swing.JScrollPane();
    jTable2 = new javax.swing.JTable();
    StructureAlignment = new javax.swing.JPanel();
    jLabel7 = new javax.swing.JLabel();
    jScrollPane6 = new javax.swing.JScrollPane();
    jTextArea1 = new javax.swing.JTextArea();
    jButton4 = new javax.swing.JButton();
```

```
jScrollPane2 = new javax.swing.JScrollPane();
jTextArea2 = new javax.swing.JTextArea();
LitSearch = new javax.swing.JPanel();
jTextField3 = new javax.swing.JTextField();
jButton2 = new javax.swing.JButton();
jScrollPane3 = new javax.swing.JScrollPane();
jTable3 = new javax.swing.JTable();
Citation = new javax.swing.JPanel();
jTextField4 = new javax.swing.JTextField();
jButton3 = new javax.swing.JButton();
jLabel5 = new javax.swing.JLabel();
jScrollPane4 = new javax.swing.JScrollPane();
jTextArea3 = new javax.swing.JTextArea();
jLabel10 = new javax.swing.JLabel();
jScrollPane5 = new javax.swing.JScrollPane();
jTextArea4 = new javax.swing.JTextArea();
jLabel11 = new javax.swing.JLabel();
jScrollPane8 = new javax.swing.JScrollPane();
jTextArea5 = new javax.swing.JTextArea();
UserAccount = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jTextField5 = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jTextField6 = new javax.swing.JTextField();
jLabel8 = new javax.swing.JLabel();
jTextField7 = new javax.swing.JTextField();
jButton7 = new javax.swing.JButton();
jButton8 = new javax.swing.JButton();
jLabel9 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Protein Toolkit");
setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
setForeground(new java.awt.Color(0, 0, 0));
setSize(new java.awt.Dimension(720, 480));

jPanel1.setBackground(new java.awt.Color(204, 204, 255));
jPanel1.setPreferredSize(new java.awt.Dimension(50, 50));

jLabel2.setText("Default Email ID");

jLabel3.setText("Default License Key");
```

```
        jLabel4.setText("Unregistered");

        jLabel15.setFont(new java.awt.Font("Times New Roman", 0, 24)); // NOI18N
        jLabel15.setText("Protein Research Toolkit");

        jButton6.setText("Activate Software");
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel15)
                .addGap(12, 12, 12)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jLabel2))
                    .addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton6)
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                    .addGroup(jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                    .addComponent(jLabel15, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                        .addComponent(jLabel2)
                        .addComponent(jLabel4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 7,
Short.MAX_VALUE)
                        .addComponent(jLabel3)))
                .addGap(0, 0, Short.MAX_VALUE))
            .addComponent(jButton6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap())
    );

    jTabbedPane1.setToolTipText("");
    jTabbedPane1.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

    UniProtSearch.setBackground(new java.awt.Color(255, 204, 204));
    UniProtSearch.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

    jTextField1.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
    jTextField1.setText("Search Protein here...");
    jTextField1.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jTextField1MouseClicked(evt);
        }
    });
    jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyPressed(java.awt.event.KeyEvent evt) {
            jTextField1KeyPressed(evt);
        }
    });

    jButton1.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
    jButton1.setText("Search");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton1ActionPerformed(evt);
        }
    });

    jTable1.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {

        },
        new String [] {
            "Protein ID", "Entry Name", "Length", "Organism", "Gene Name", "FASTA
Sequence"
        }
    ) {
        boolean[] canEdit = new boolean [] {
            false, false, false, false, false, false
        };

        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    });
    jTable1.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_ALL_COLUMNS);
    jTable1.setColumnSelectionAllowed(true);
    jTable1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
    jTable1.setName("UniProt Results"); // NOI18N
    jTable1.setRowHeight(25);
    jTable1.setShowGrid(true);
    jScrollPane1.setViewportView(jTable1);

jTable1.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelection
Model.MULTIPLE_INTERVAL_SELECTION);

    javax.swing.GroupLayout UniProtSearchLayout = new
javax.swing.GroupLayout(UniProtSearch);
    UniProtSearch.setLayout(UniProtSearchLayout);
    UniProtSearchLayout.setHorizontalGroup(

UniProtSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(UniProtSearchLayout.createSequentialGroup()
            .addContainerGap()

.addGroup(UniProtSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
```

```
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
1044, Short.MAX_VALUE)
                .addGroup(UniProtSearchLayout.createSequentialGroup()
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
441, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jButton1)
                    .addGap(0, 0, Short.MAX_VALUE)))
            .addContainerGap())
        );
        UniProtSearchLayout.setVerticalGroup(

UniProtSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(UniProtSearchLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(UniProtSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)
                .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 35,
Short.MAX_VALUE)
                .addComponent(jTextField1))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 691,
Short.MAX_VALUE)
                .addContainerGap())
        );

        jTabbedPane1.addTab("Protein Sequence Search", UniProtSearch);

        PDBSearch.setBackground(new java.awt.Color(255, 204, 204));

        jTextField2.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
        jTextField2.setText("Search Protein here...");
        jTextField2.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTextField2MouseClicked(evt);
            }
        });
        jTextField2.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextField2KeyPressed(evt);
            }
        });
```

```
        jButton5.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
        jButton5.setText("Search");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton5ActionPerformed(evt);
            }
        });

        jTable2.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
        jTable2.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Protein ID", "Entry Name", "Method", "Visualization", "Structure Alignment"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        jTable2.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_ALL_COLUMNS);
        jTable2.setCellSelectionEnabled(true);
        jTable2.setRowHeight(25);
        jTable2.setShowGrid(true);
        jTable2.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTable2MouseClicked(evt);
            }
        });
        jScrollPane7.setViewportView(jTable2);

jTable2.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelection
Model.MULTIPLE_INTERVAL_SELECTION);

        javax.swing.GroupLayout PDBSearchLayout = new
javax.swing.GroupLayout(PDBSearch);
        PDBSearch.setLayout(PDBSearchLayout);
        PDBSearchLayout.setHorizontalGroup(
```

```
PDBSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(PDBSearchLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(PDBSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jScrollPane7, javax.swing.GroupLayout.DEFAULT_SIZE,
1044, Short.MAX_VALUE)
                    .addGroup(PDBSearchLayout.createSequentialGroup()
                        .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
439, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton5)
                        .addGap(0, 0, Short.MAX_VALUE)))
                .addContainerGap())
        );
        PDBSearchLayout.setVerticalGroup(

PDBSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(PDBSearchLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(PDBSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jButton5, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane7, javax.swing.GroupLayout.DEFAULT_SIZE, 691,
Short.MAX_VALUE)
                .addContainerGap())
        );

        jTabbedPane1.addTab("Structure Visualization", PDBSearch);

        StructureAlignment.setBackground(new java.awt.Color(255, 204, 204));

        jLabel7.setText("Enter List of PDB IDs:");

        jTextArea1.setColumns(20);
        jTextArea1.setRows(5);
        jScrollPane6.setViewportView(jTextArea1);
```

```
        jButton4.setText("Align");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });

        jTextArea2.setColumns(20);
        jTextArea2.setRows(5);
        jScrollPane2.setViewportView(jTextArea2);

        javax.swing.GroupLayout StructureAlignmentLayout = new
javax.swing.GroupLayout(StructureAlignment);
        StructureAlignment.setLayout(StructureAlignmentLayout);
        StructureAlignmentLayout.setHorizontalGroup(

StructureAlignmentLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
            .addGroup(StructureAlignmentLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(StructureAlignmentLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jScrollPane2)
                    .addGroup(StructureAlignmentLayout.createSequentialGroup()

.addGroup(StructureAlignmentLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
                            .addComponent(jLabel7, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jScrollPane6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(0, 691, Short.MAX_VALUE)))
                .addContainerGap())
        );
        StructureAlignmentLayout.setVerticalGroup(

StructureAlignmentLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
            .addGroup(StructureAlignmentLayout.createSequentialGroup()
```

```
            .addContainerGap()

.addGroup(StructureAlignmentLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
                .addGroup(StructureAlignmentLayout.createSequentialGroup()
                    .addComponent(jLabel7)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jButton4))
                .addComponent(jScrollPane6, javax.swing.GroupLayout.PREFERRED_SIZE,
150, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 570,
Short.MAX_VALUE)
            .addContainerGap())
        );

        jTabbedPane1.addTab("Structural Alignment", StructureAlignment);

        LitSearch.setBackground(new java.awt.Color(255, 204, 204));

        jTextField3.setText("Enter a Query...");
        jTextField3.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTextField3MouseClicked(evt);
            }
        });
        jTextField3.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextField3KeyPressed(evt);
            }
        });

        jButton2.setText("Search Literature");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jTable3.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
```

```
            new String [] {
                "DOI ID", "Title", "Authors", "Year", "No. of times cited", "Link to Article"
            }
        ) {
            Class[] types = new Class [] {
                java.lang.Object.class, java.lang.Object.class, java.lang.String.class,
java.lang.Object.class, java.lang.Object.class, java.lang.Object.class
            };
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false
            };

            public Class getColumnClass(int columnIndex) {
                return types [columnIndex];
            }

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        jTable3.setAutoResizeMode(javax.swing.JTable.AUTO_RESIZE_ALL_COLUMNS);
        jTable3.setCellSelectionEnabled(true);
        jScrollPane3.setViewportView(jTable3);

        javax.swing.GroupLayout LitSearchLayout = new
javax.swing.GroupLayout(LitSearch);
        LitSearch.setLayout(LitSearchLayout);
        LitSearchLayout.setHorizontalGroup(

LitSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(LitSearchLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(LitSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
                    .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE,
1044, Short.MAX_VALUE)
                    .addGroup(LitSearchLayout.createSequentialGroup()
                        .addGap(6, 6, 6)
                        .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
436, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton2)
                        .addGap(0, 0, Short.MAX_VALUE)))
```

```
            .addContainerGap())
        );
        LitSearchLayout.setVerticalGroup(

LitSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(LitSearchLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(LitSearchLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
                    .addComponent(jTextField3)
                    .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE, 35,
Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT_SIZE, 691,
Short.MAX_VALUE)
                .addContainerGap())
        );

        jTabbedPane1.addTab("Literature Search", LitSearch);
        LitSearch.getAccessibleContext().setAccessibleName("");

        Citation.setBackground(new java.awt.Color(255, 204, 204));

        jTextField4.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
        jTextField4.setText("Enter DOI ID...");
        jTextField4.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTextField4MouseClicked(evt);
            }
        });
        jTextField4.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextField4KeyPressed(evt);
            }
        });

        jButton3.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N
        jButton3.setText("Search");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });
```

```
        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        jLabel5.setText("APA Format:");

        jTextArea3.setColumns(20);
        jTextArea3.setRows(5);
        jScrollPane4.setViewportView(jTextArea3);

        jLabel10.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        jLabel10.setText("MLA Format:");

        jTextArea4.setColumns(20);
        jTextArea4.setRows(5);
        jScrollPane5.setViewportView(jTextArea4);

        jLabel11.setFont(new java.awt.Font("Times New Roman", 1, 18)); // NOI18N
        jLabel11.setText("Harvard Format:");

        jTextArea5.setColumns(20);
        jTextArea5.setRows(5);
        jScrollPane8.setViewportView(jTextArea5);

        javax.swing.GroupLayout CitationLayout = new javax.swing.GroupLayout(Citation);
        Citation.setLayout(CitationLayout);
        CitationLayout.setHorizontalGroup(

CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(CitationLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                    .addGroup(CitationLayout.createSequentialGroup()
                        .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
441, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton3))
                    .addGroup(CitationLayout.createSequentialGroup()

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                            .addComponent(jLabel11, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                    .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel10, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(18, 18, 18)

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAIL
ING, false)
                    .addComponent(jScrollPane5,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 862, Short.MAX_VALUE)
                    .addComponent(jScrollPane4,
javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane8))))
            .addContainerGap(38, Short.MAX_VALUE))
        );
        CitationLayout.setVerticalGroup(

CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(CitationLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                    .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 48, Short.MAX_VALUE)

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                    .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT_SIZE, 120,
Short.MAX_VALUE))
                .addGap(18, 18, 18)

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                    .addComponent(jLabel10, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT_SIZE, 120,
Short.MAX_VALUE))
```

```
            .addGap(18, 18, 18)

.addGroup(CitationLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                .addComponent(jLabel11, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jScrollPane8, javax.swing.GroupLayout.PREFERRED_SIZE,
120, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(259, 259, 259))
        );

        jTabbedPane1.addTab("Generate Citation", Citation);

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jLabel1.setText("Email ID:");

        jTextField5.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

        jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jLabel6.setText("Password:");

        jTextField6.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

        jLabel8.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jLabel8.setText("License Key:");

        jTextField7.setEditable(false);
        jTextField7.setFont(new java.awt.Font("Times New Roman", 0, 14)); // NOI18N

        jButton7.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jButton7.setText("Login");
        jButton7.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton7ActionPerformed(evt);
            }
        });

        jButton8.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jButton8.setText("Register");
        jButton8.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton8ActionPerformed(evt);
            }
        });
```

```
        jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
        jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel9.setText(" ");

        javax.swing.GroupLayout UserAccountLayout = new
javax.swing.GroupLayout(UserAccount);
        UserAccount.setLayout(UserAccountLayout);
        UserAccountLayout.setHorizontalGroup(

UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
UserAccountLayout.createSequentialGroup()
                .addContainerGap(369, Short.MAX_VALUE)

.addGroup(UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING, false)
                    .addGroup(UserAccountLayout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(18, 18, 18)
                        .addComponent(jTextField5))
                    .addGroup(UserAccountLayout.createSequentialGroup()
                        .addComponent(jLabel6)
                        .addGap(18, 18, 18)
                        .addComponent(jTextField6))
                    .addGroup(UserAccountLayout.createSequentialGroup()
                        .addComponent(jLabel8)
                        .addGap(18, 18, 18)
                        .addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED_SIZE,
262, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
UserAccountLayout.createSequentialGroup()
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
26, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jButton8)
                        .addGap(145, 145, 145)
                        .addComponent(jButton7)
                        .addGap(27, 27, 27))
                    .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 360,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(327, 327, 327))
        );
        UserAccountLayout.setVerticalGroup(
```

```
UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(UserAccountLayout.createSequentialGroup()
                .addGap(157, 157, 157)

.addGroup(UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel1)
                    .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel6)
                    .addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jLabel8)
                    .addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

.addGroup(UserAccountLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
                    .addComponent(jButton7)
                    .addComponent(jButton8))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(376, Short.MAX_VALUE))
        );

        jTabbedPane1.addTab("User Account", UserAccount);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
```

```
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, 1061,
Short.MAX_VALUE)
                .addComponent(jTabbedPane1))
            .addContainerGap())
      );
      layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jTabbedPane1)
            .addContainerGap())
      );

      jTabbedPane1.getAccessibleContext().setAccessibleName("Protein Structure Search");

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

   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      pd.processDialog(() -> {
         try {
            uni.dispUniProt();
         } catch (UnsupportedEncodingException ex) {
            Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
         }
      });
   }

   private void jTextField1MouseClicked(java.awt.event.MouseEvent evt) {
      if(jTextField1.getText().equals("Search Protein here...")) {
         jTextField1.setText("");
      }
   }

   private void jTextField1KeyPressed(java.awt.event.KeyEvent evt) {
```

```
    if (evt.getKeyCode() == evt.VK_ENTER) {
        pd.processDialog(() -> {
            try {
                uni.dispUniProt();
            } catch (UnsupportedEncodingException ex) {
                Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
            }
        });
    }
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    pd.processDialog(() -> {
        try {
            prntStructData();
        } catch (UnsupportedEncodingException ex) {
            Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
        }
    });
}

private void jTextField2MouseClicked(java.awt.event.MouseEvent evt) {
    if(jTextField2.getText().equals("Search Protein here...")) {
        jTextField2.setText("");
    }
}

private void jTextField2KeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == evt.VK_ENTER) {
        pd.processDialog(() -> {
            try {
                prntStructData();
            } catch (UnsupportedEncodingException ex) {
                Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
            }
        });
    }
}

private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
    int row = jTable2.rowAtPoint(evt.getPoint());
    int col = jTable2.columnAtPoint(evt.getPoint());
    Object value = jTable2.getValueAt(row, 0);
    if (row >= 0 && col == 3) {
```

```
        pd.processDialog(() -> vs.VisStruct(value.toString()));
    }
    if (row >= 0 && col == 4) {
        pd.processDialog(() -> jTextArea1.append(value.toString()+"\n"));
    }
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    pd.processDialog(() -> sa.StructAli(jTextArea1.getText()));
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    pd.processDialog(() -> {
        try {
            as.ArticleSearch(jTextField3.getText());
        } catch (Exception ex) {
            Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
        }
    });
}

private void jTextField3KeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == evt.VK_ENTER) {
        pd.processDialog(() -> {
            try {
                as.ArticleSearch(jTextField3.getText());
            } catch (Exception ex) {
                Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);
            }
        });
    }
}

private void jTextField4MouseClicked(java.awt.event.MouseEvent evt) {
    if(jTextField2.getText().equals("Enter DOI ID...")) {
        jTextField2.setText("");
    }
}

private void jTextField4KeyPressed(java.awt.event.KeyEvent evt) {
    if (evt.getKeyCode() == evt.VK_ENTER) {
        pd.processDialog(() -> gc.generateArticleCitation(jTextField4.getText()));
    }
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextArea3.setText("");
    jTextArea4.setText("");
    jTextArea5.setText("");
    pd.processDialog(() -> gc.generateArticleCitation(jTextField4.getText()));
}

private void jTextField3MouseClicked(java.awt.event.MouseEvent evt) {
    if(jTextField3.getText().equals("Enter a Query...")) {
        jTextField3.setText("");
    }
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    jTabbedPane1.setSelectedIndex(5);
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    pd.processDialog(() -> st.register());
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    pd.processDialog(() -> st.login());
}

/**
 * @param args the command line arguments
 * @throws javax.swing.UnsupportedLookAndFeelException
 */
public static void main(String args[]) throws UnsupportedLookAndFeelException {
    UIManager.setLookAndFeel(new FlatMacLightLaf());

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new MainFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JPanel Citation;
private javax.swing.JPanel LitSearch;
```

```
private javax.swing.JPanel PDBSearch;
private javax.swing.JPanel StructureAlignment;
private javax.swing.JPanel UniProtSearch;
private javax.swing.JPanel UserAccount;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel15;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JScrollPane jScrollPane4;
private javax.swing.JScrollPane jScrollPane5;
private javax.swing.JScrollPane jScrollPane6;
private javax.swing.JScrollPane jScrollPane7;
private javax.swing.JScrollPane jScrollPane8;
private javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JTable jTable1;
public javax.swing.JTable jTable2;
public javax.swing.JTable jTable3;
public javax.swing.JTextArea jTextArea1;
public javax.swing.JTextArea jTextArea2;
public javax.swing.JTextArea jTextArea3;
public javax.swing.JTextArea jTextArea4;
public javax.swing.JTextArea jTextArea5;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
```

```
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    public javax.swing.JTextField jTextField5;
    public javax.swing.JTextField jTextField6;
    public javax.swing.JTextField jTextField7;
    // End of variables declaration
}
```

OUTPUT:

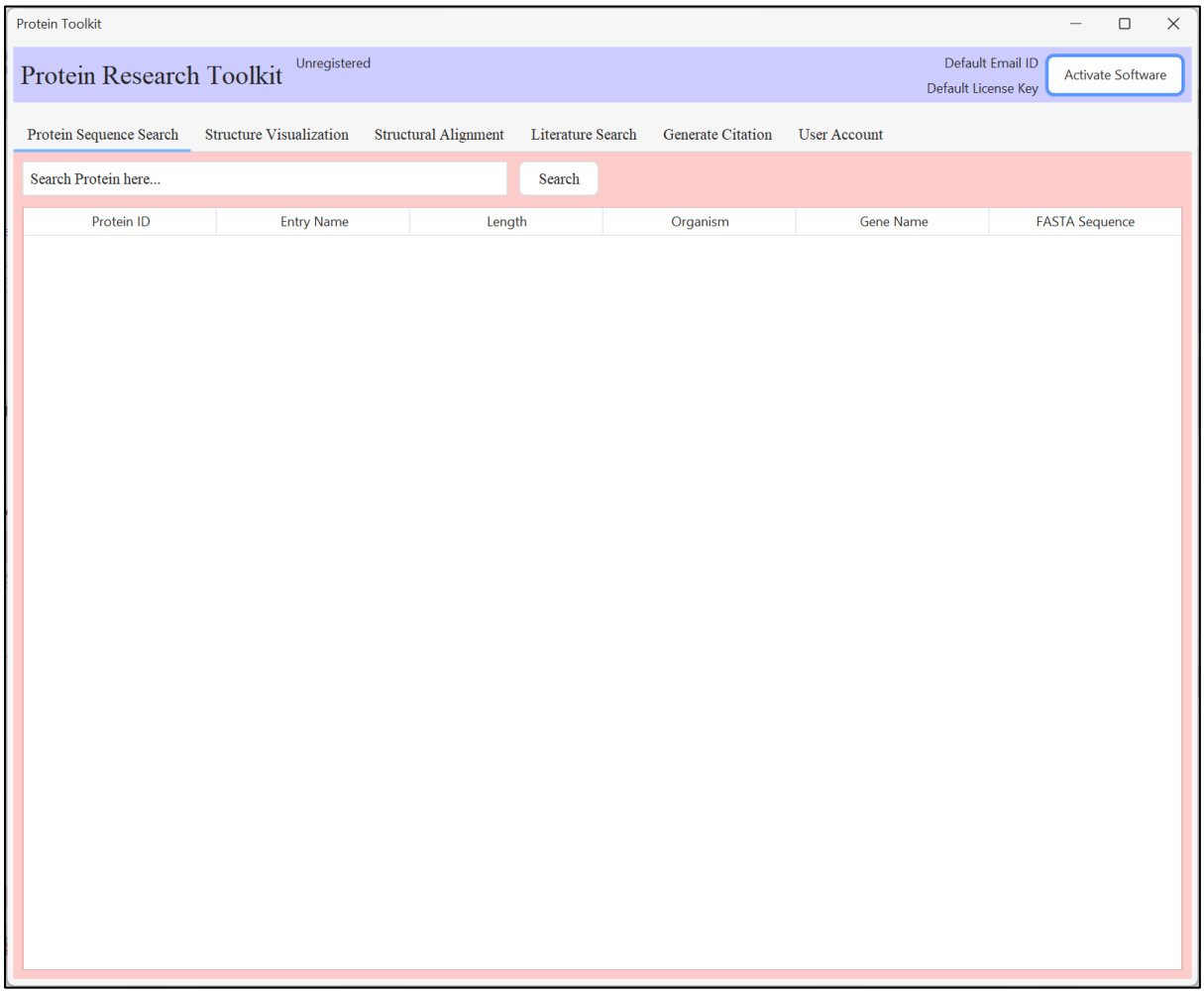


Figure 1: Homepage of Protein Research Toolkit

1. Login

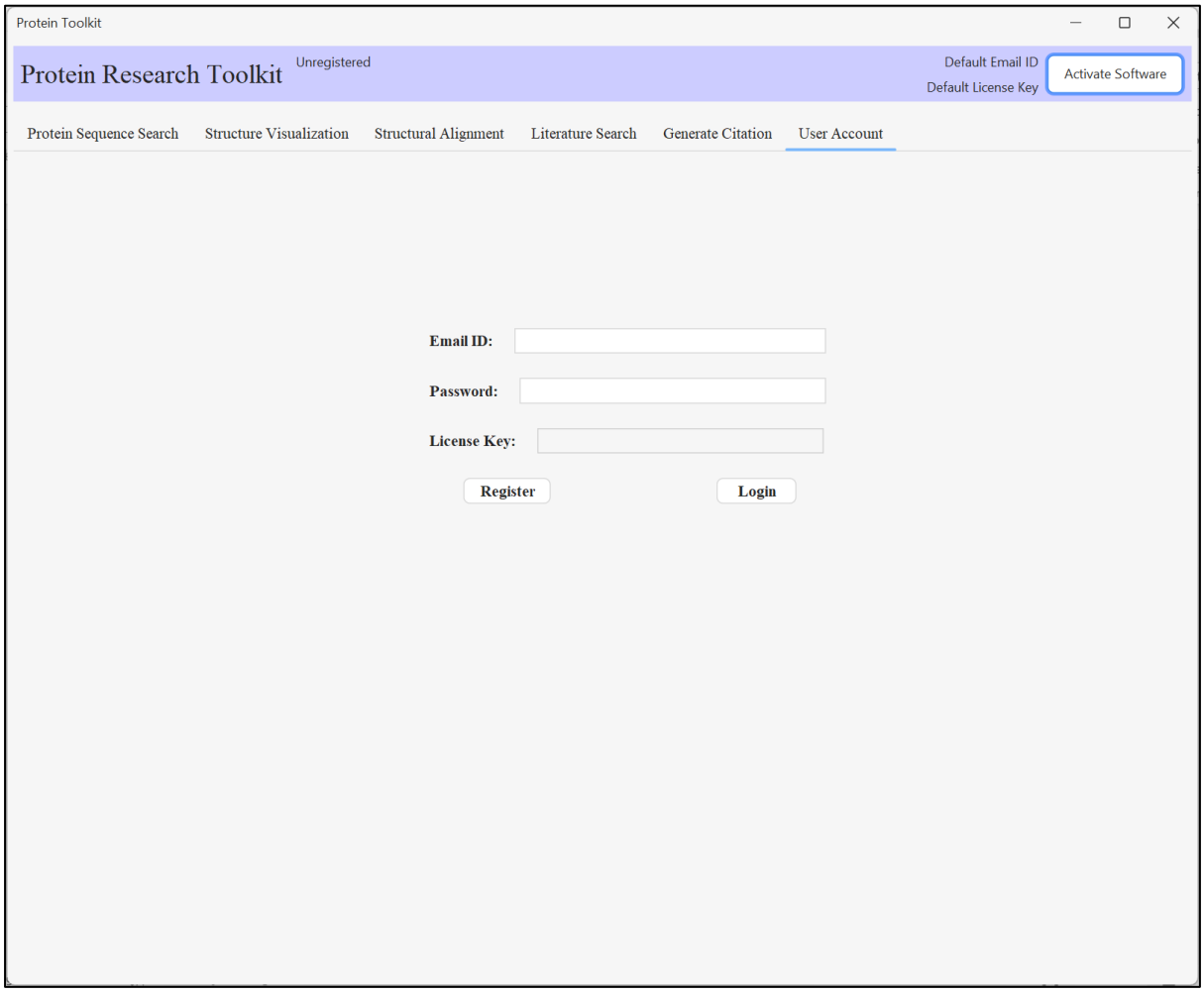


Figure 2: Login Page

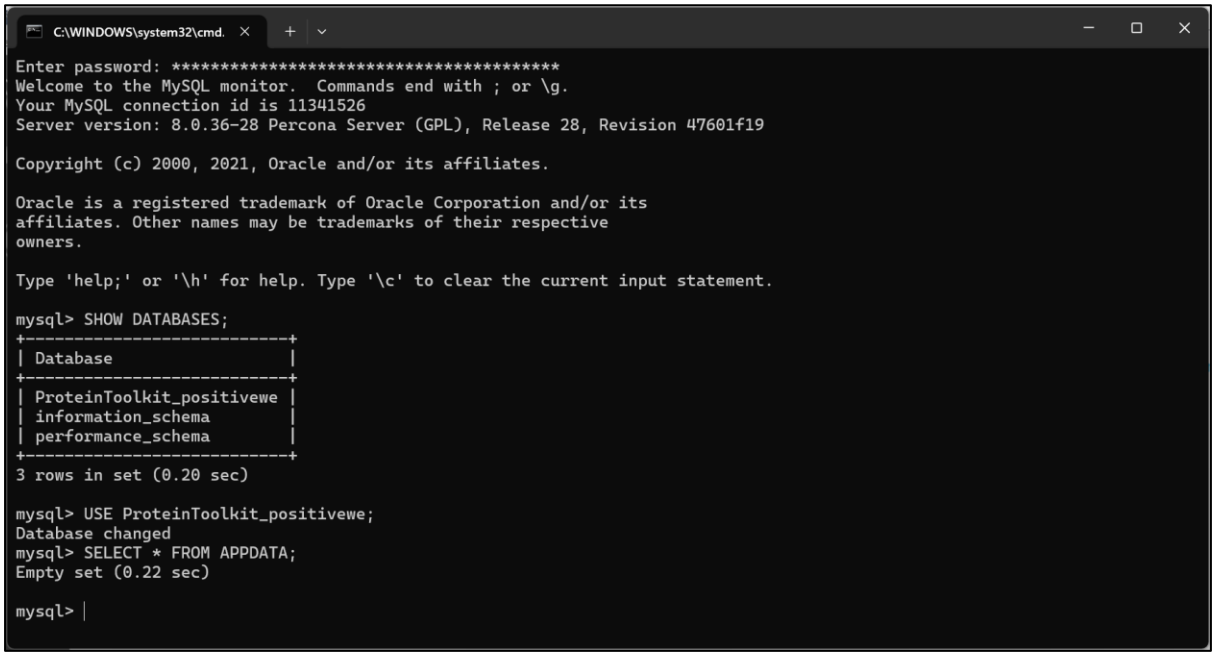


Figure 3: MySQL Command Line: Empty Dataset (before registration)

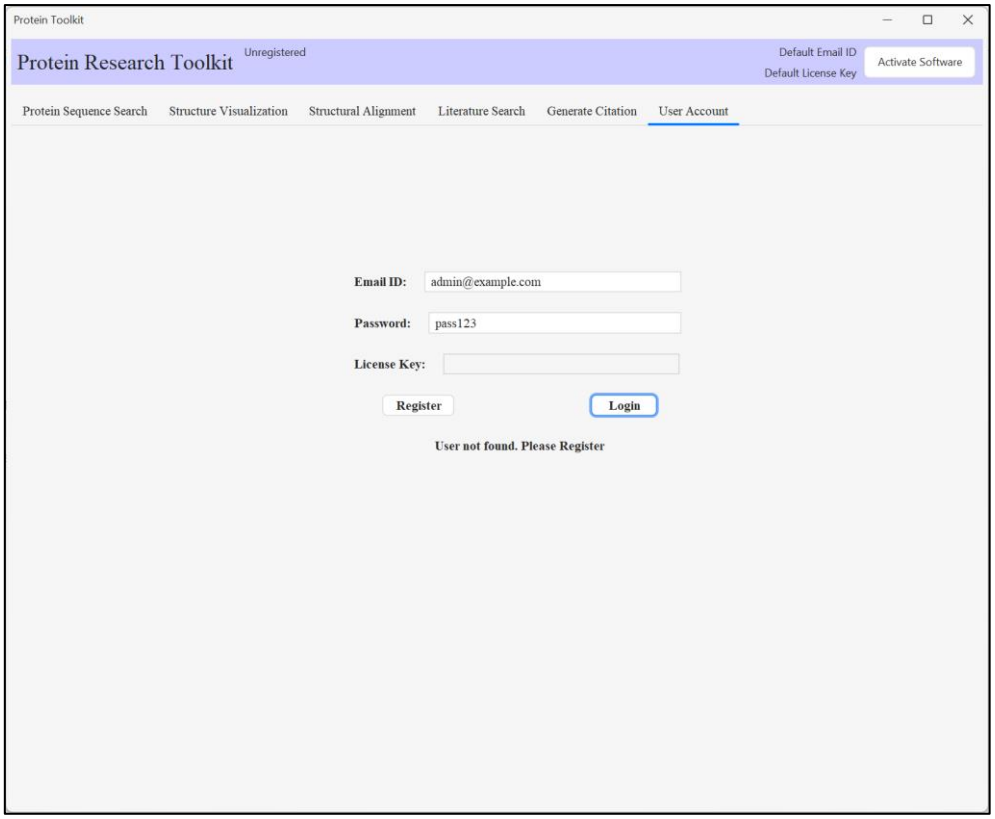


Figure 4: Login using New User Credentials: User not found (before registration)

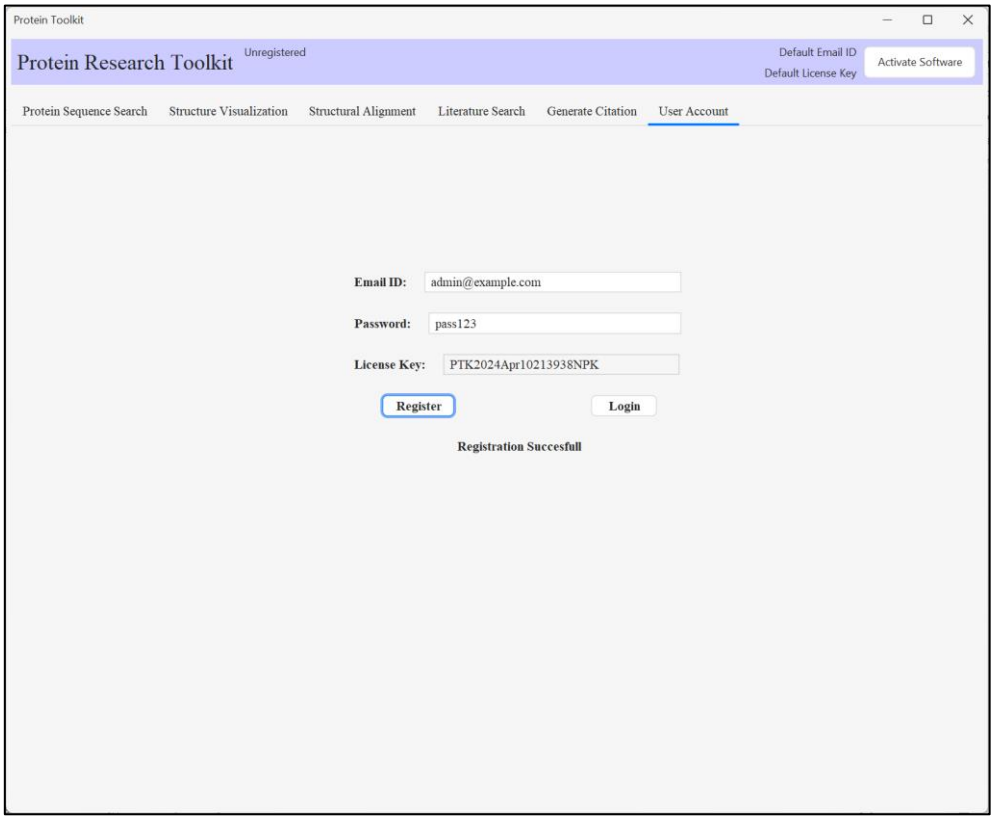


Figure 5: Registering New User and automatically generating the License Key

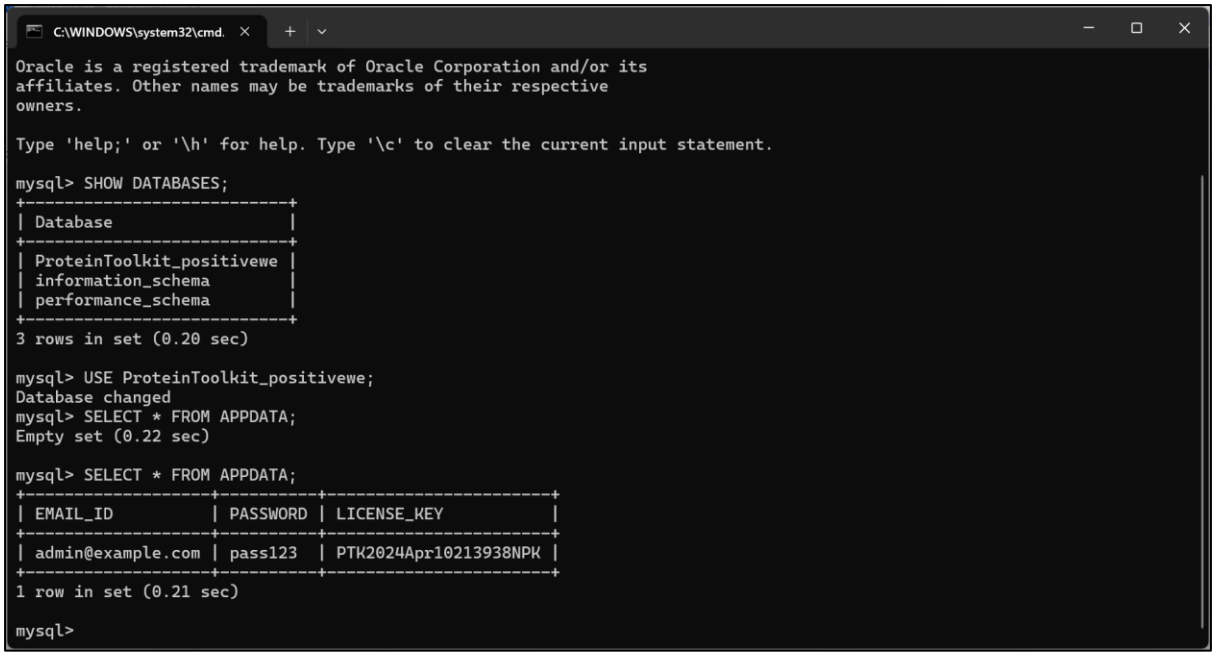


Figure 6: MySQL Command Line: Display of User Credentials (after registration)

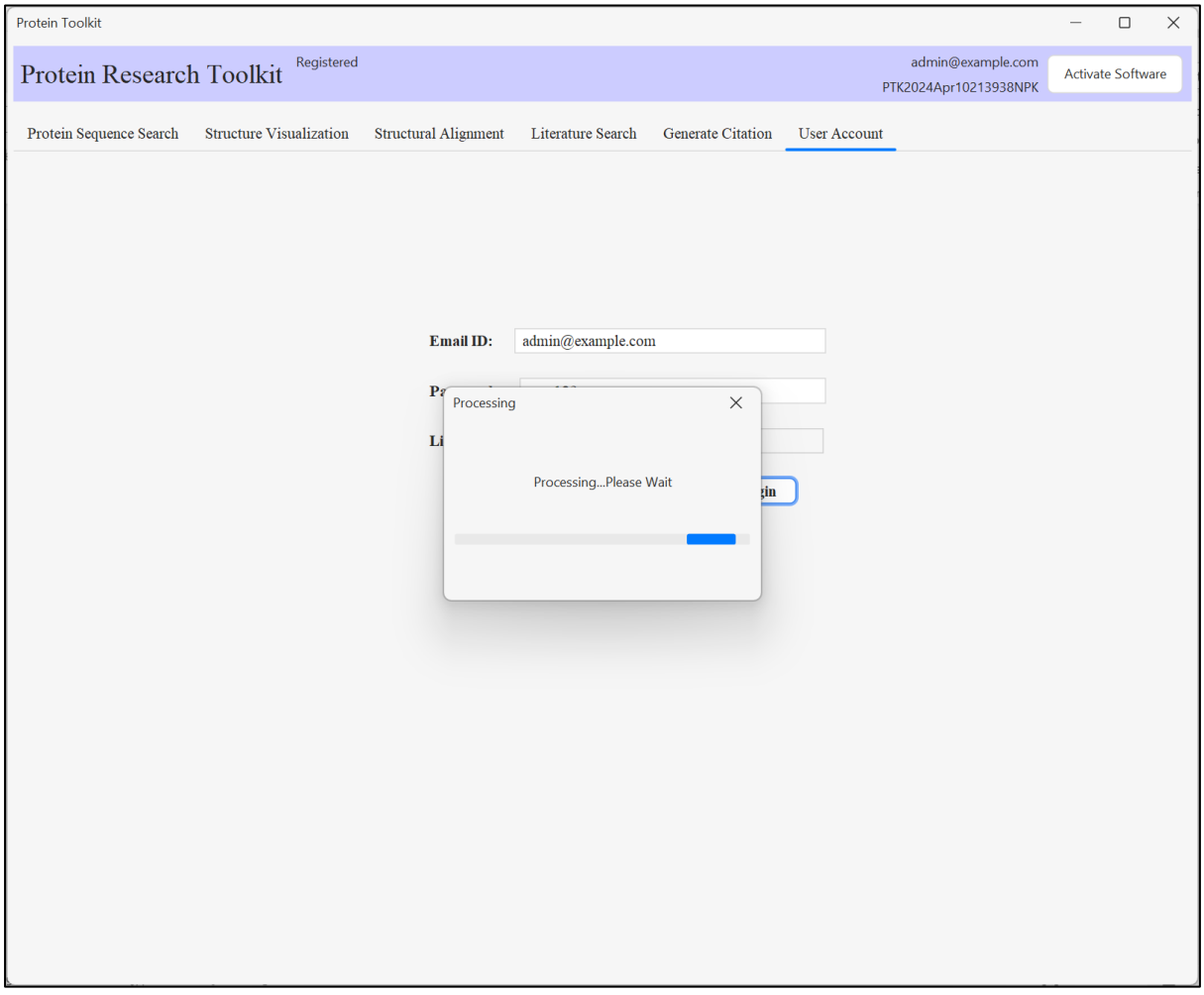


Figure 7: Login using registered user credentials

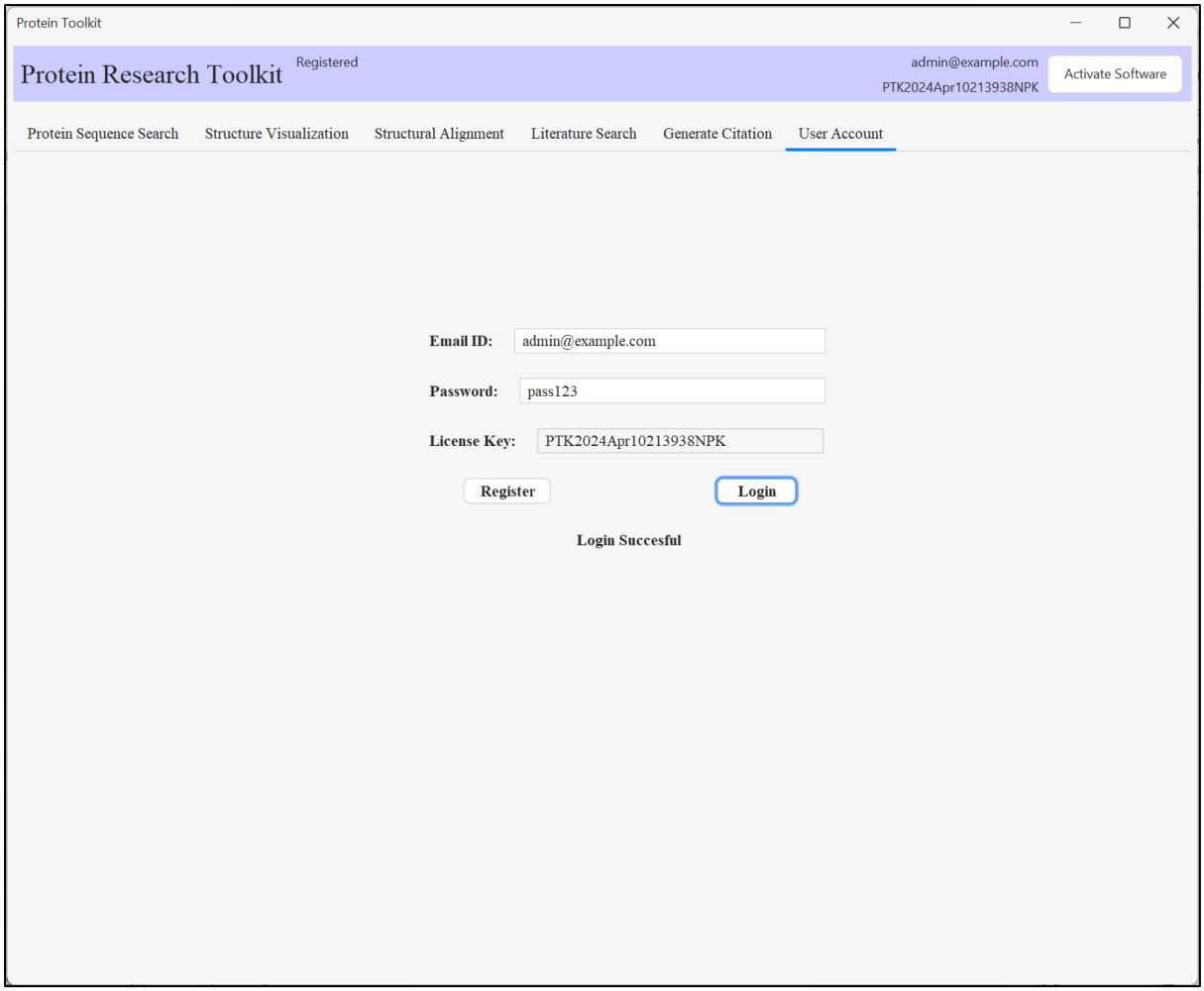


Figure 8: Successful login using correct user credentials

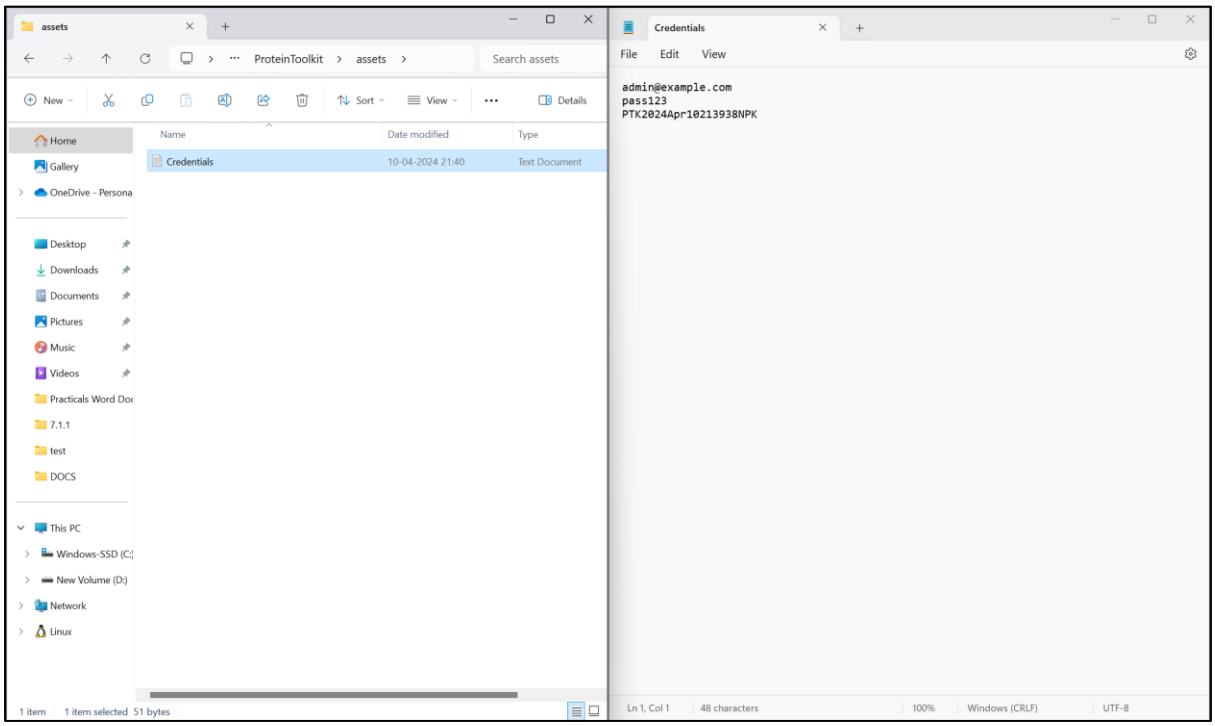


Figure 9: User credentials stored locally in ‘Credentials.txt’ file

2. Protein Sequence Search

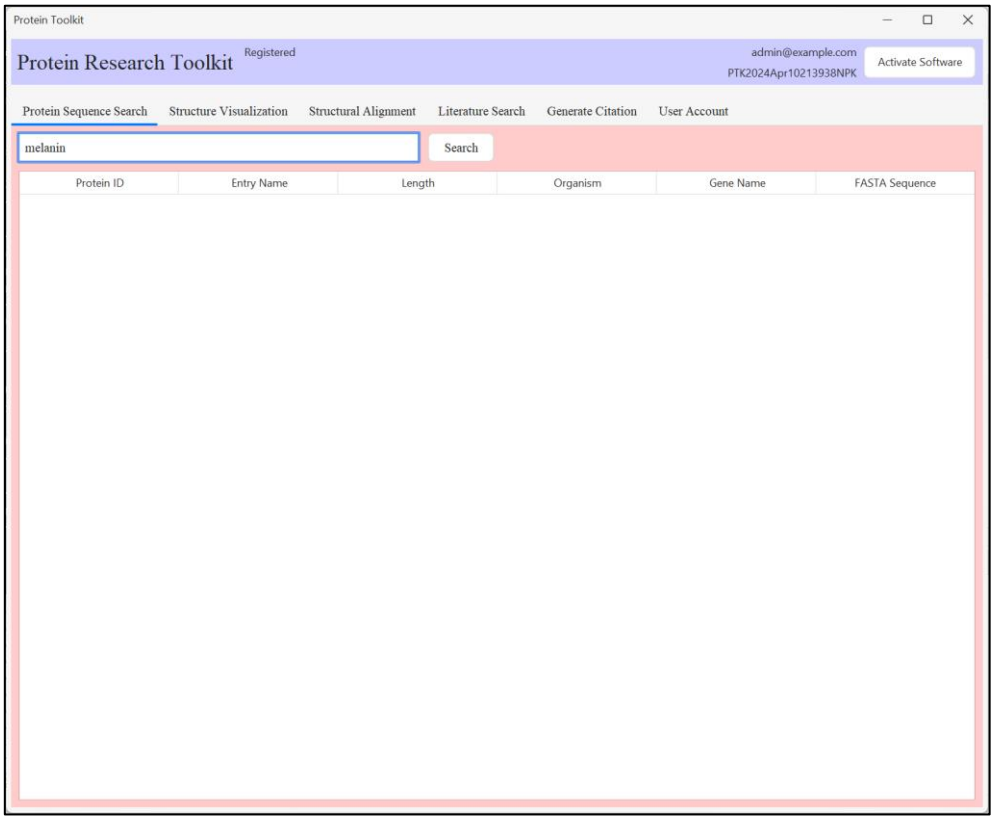


Figure 10: 'Protein Sequence Search' tool with the protein query 'melanin'

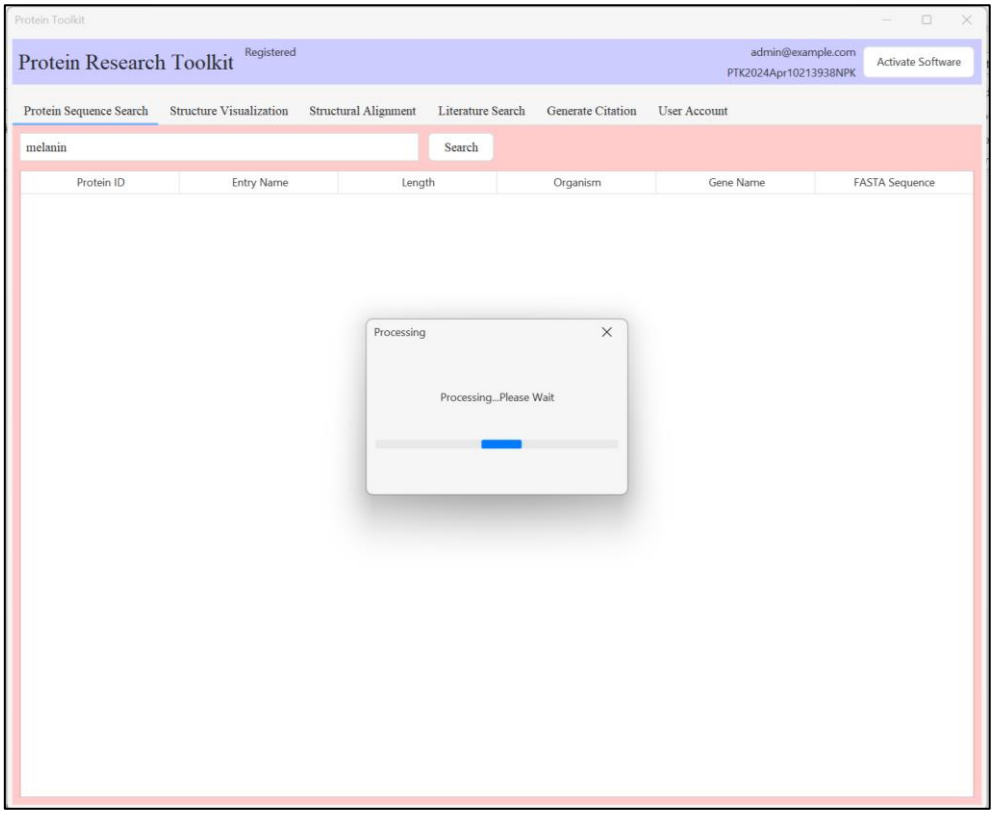


Figure 11: Searching, processing and fetching results from the UniProt database

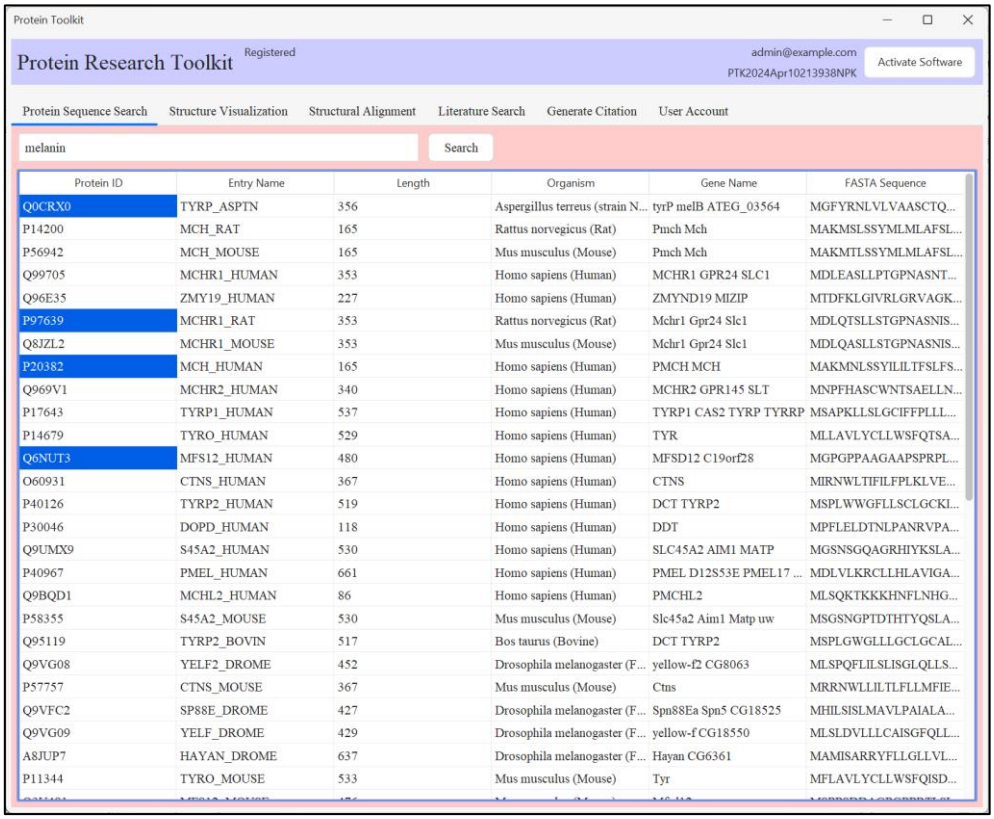


Figure 12: Results obtained in ‘Protein Sequence Search’ tool for the query ‘melanin’

3. Structure Visualization

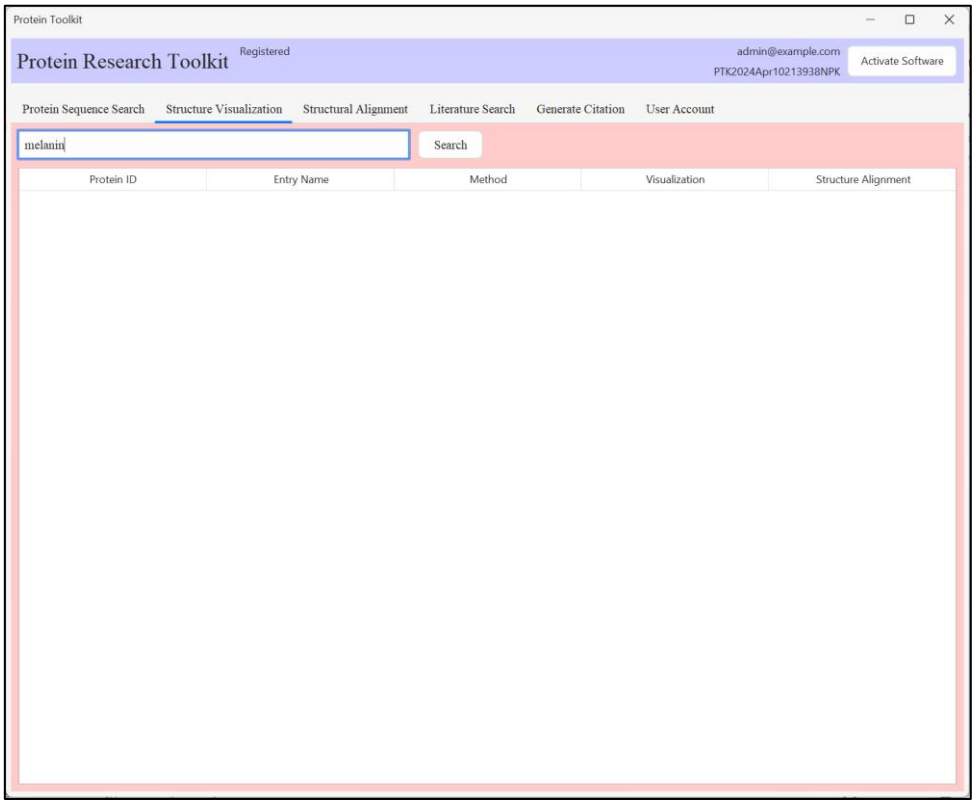


Figure 13: 'Structure Visualization' tool with the protein query 'melanin'

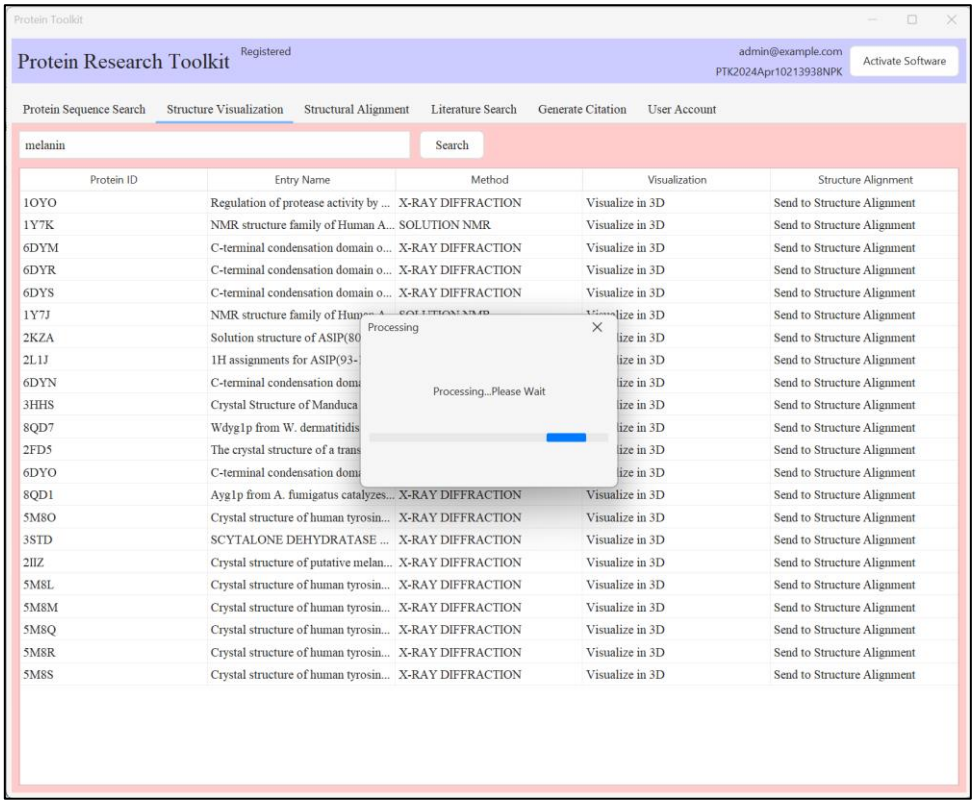


Figure 14: Searching, processing and fetching results from the Protein Data Bank (PDB) database

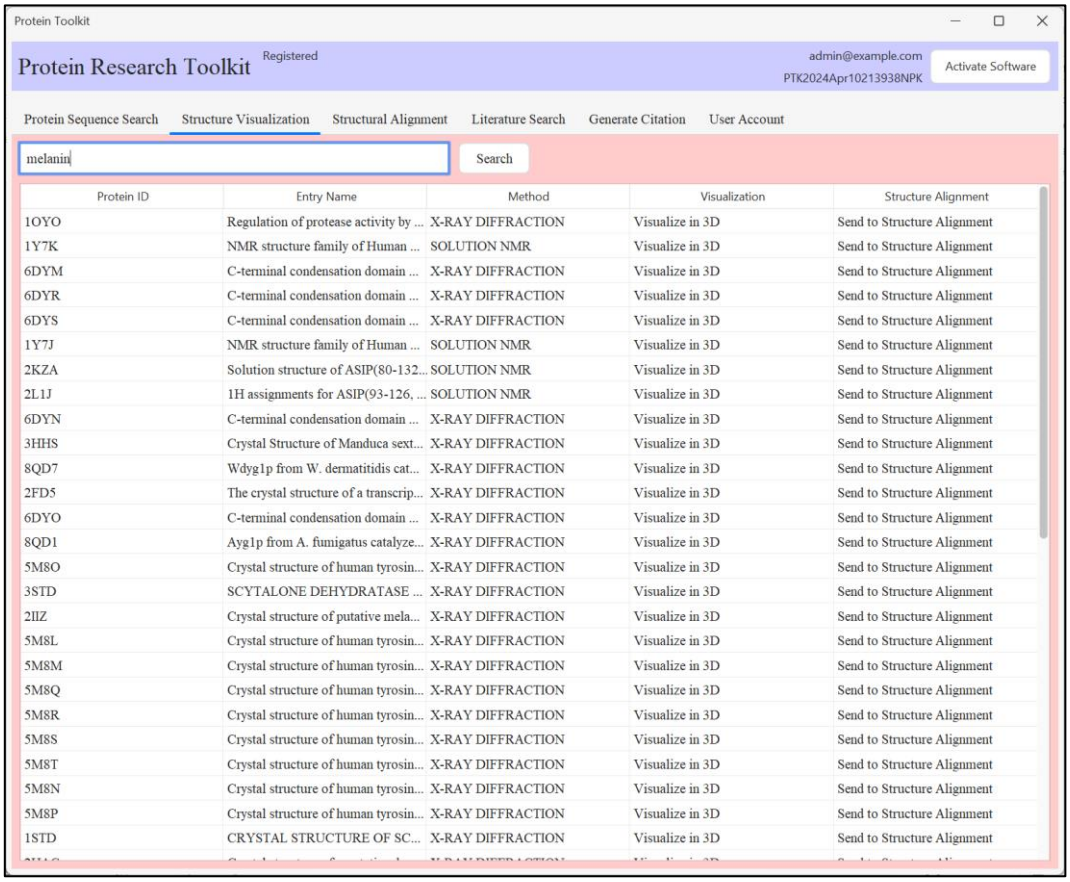


Figure 15: Results obtained in the ‘Structure Visualization’ tool for the protein query ‘melanin’

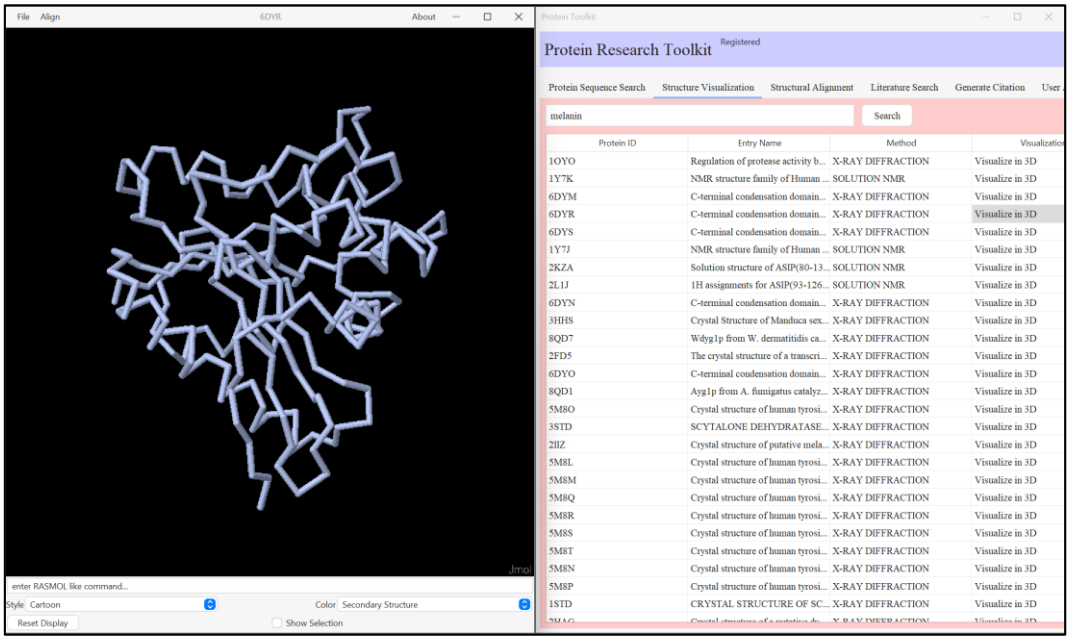


Figure 16: 3-Dimensional structure view for the protein query ‘melanin’ (PDB ID: 6DYR)

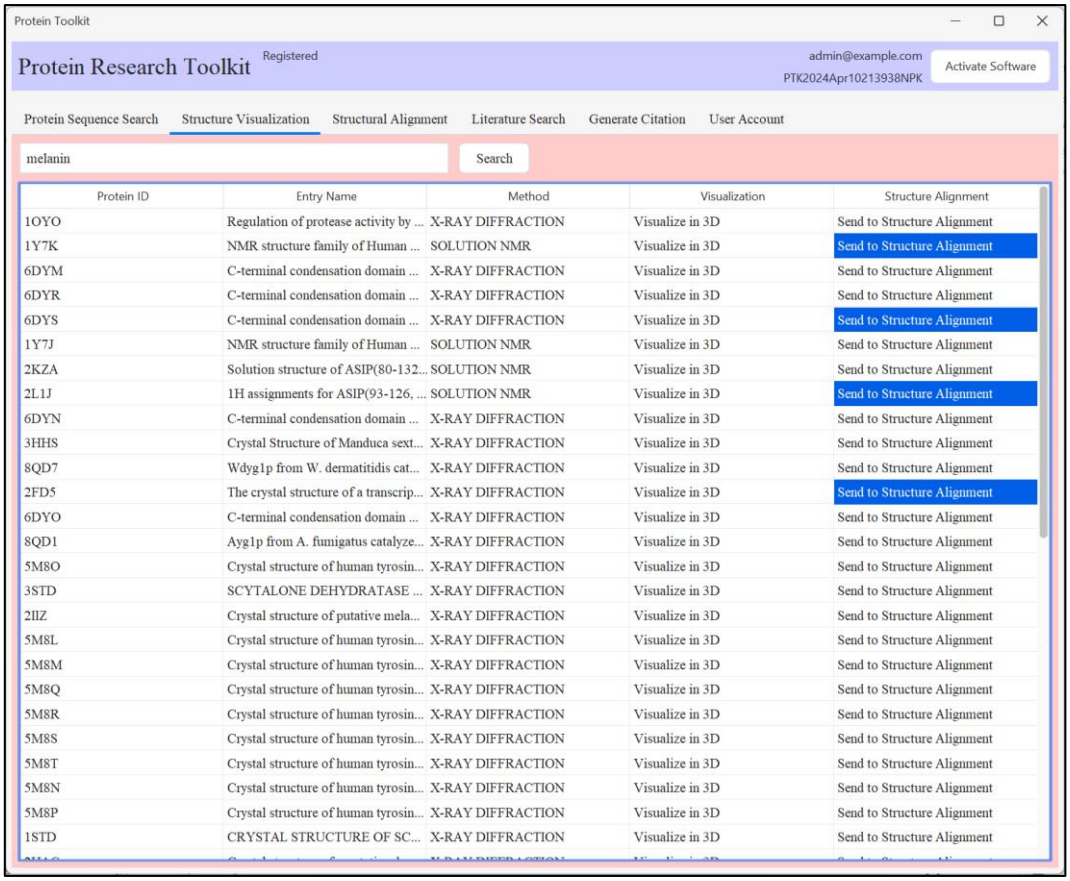


Figure 17: Protein structures with PDB IDs: 1Y7K, 6DYS, 2L1J and 2FD5 selected for further analysis in the ‘Structural Alignment’ tool

4. Structural Alignment

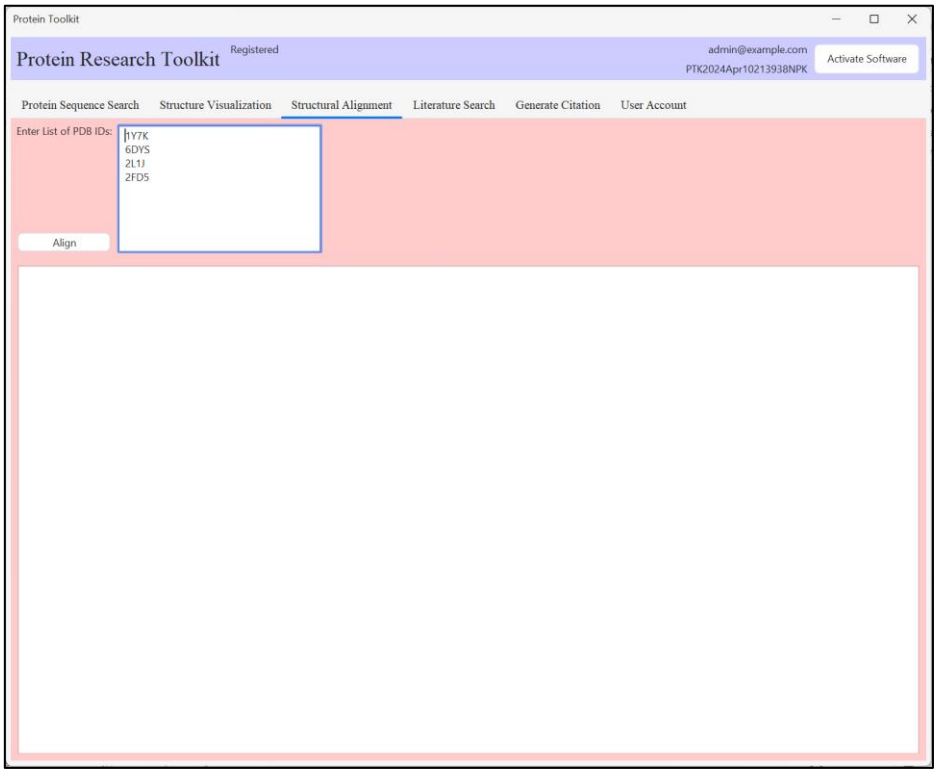


Figure 18: 'Structural Alignment' tool with the PDB IDs (1Y7K, 6DYS, 2L1J and 2FD5) of the selected protein queries of interest from the 'Structure Visualization' tool

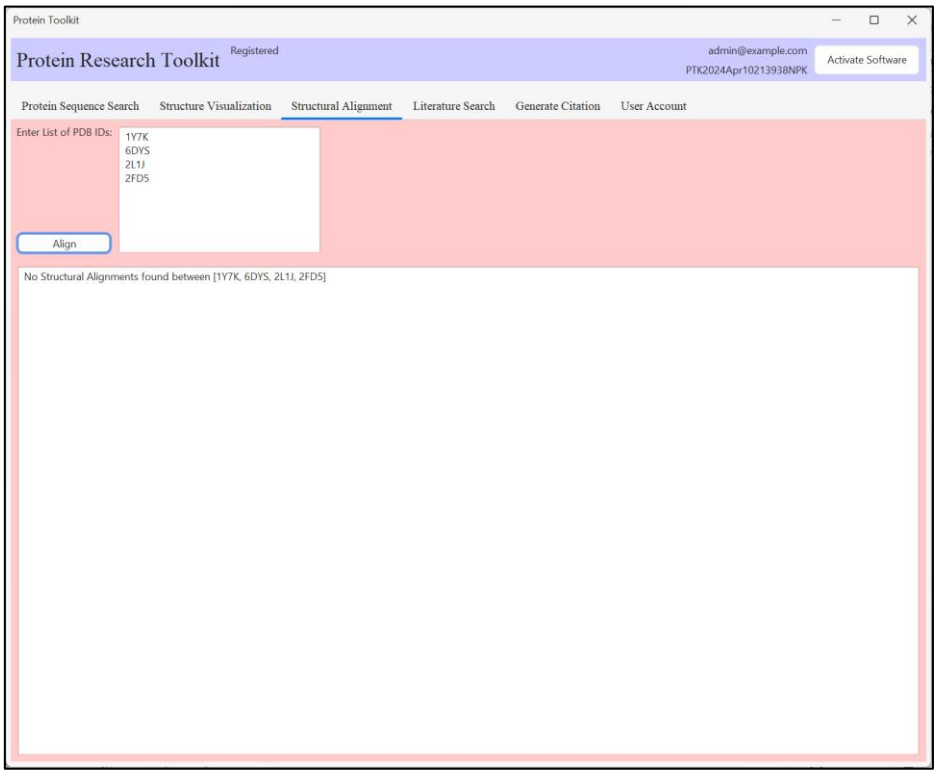


Figure 19: No Structural Alignment obtained for the selected protein queries with the PDB IDs: 1Y7K, 6DYS, 2L1J and 2FD5

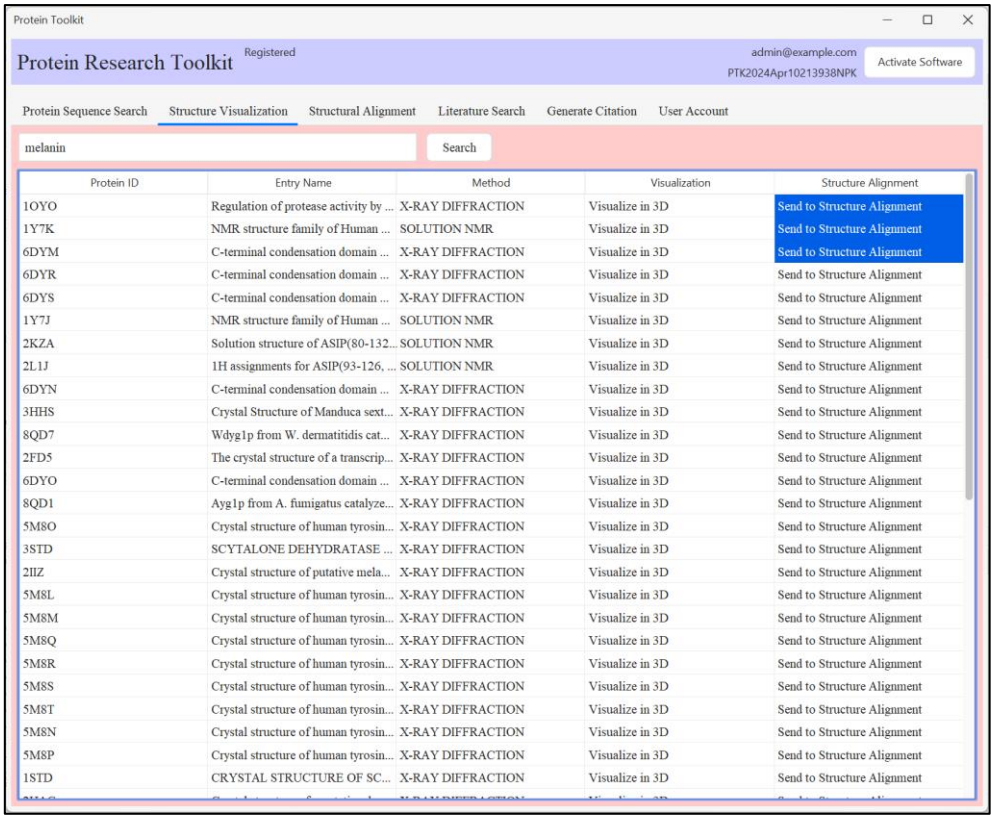


Figure 20: Protein structures with PDB IDs: 1OYO, 1Y7K and 6DYM selected for further analysis in the ‘Structural Alignment’ tool

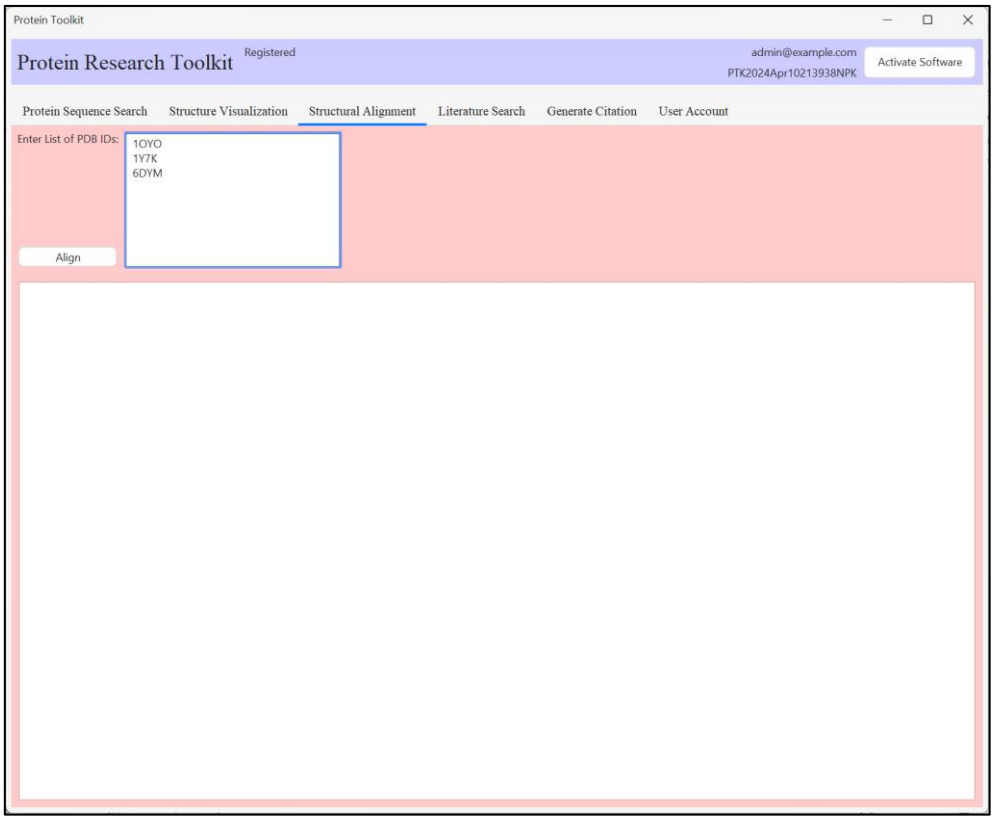


Figure 21: ‘Structural Alignment’ tool with the PDB IDs (1OYO, 1Y7K and 6DYM) of the selected protein queries of interest from the ‘Structure Visualization’ tool

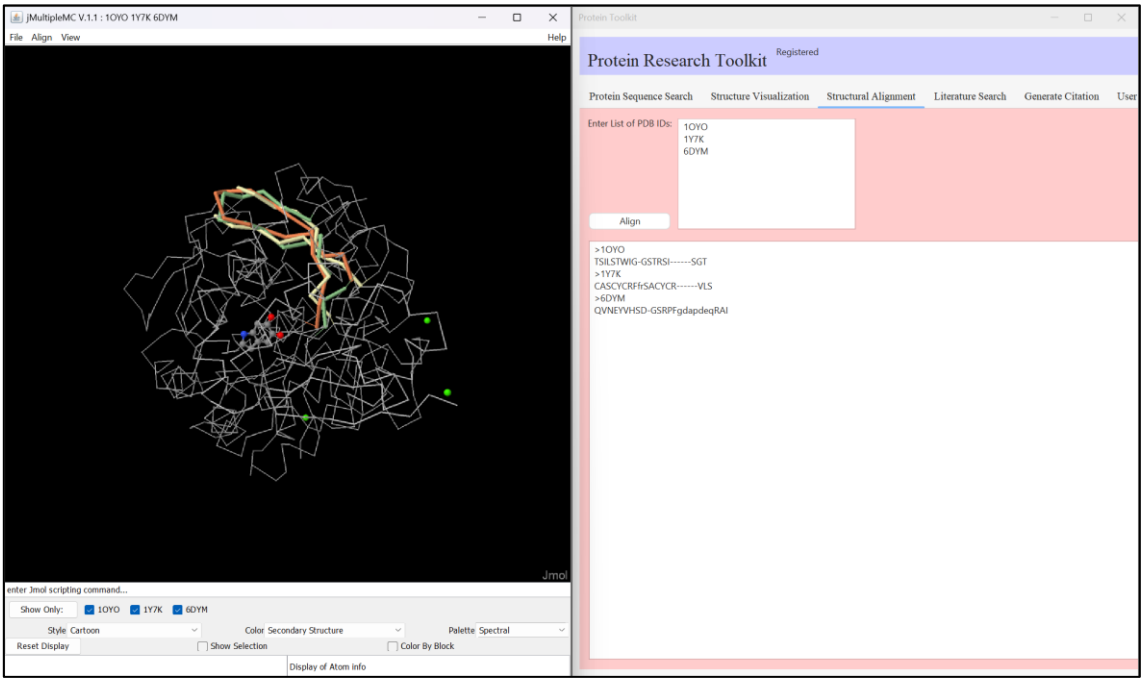


Figure 22: View of the aligned structures of the protein queries with the PDB IDs: 1OYO, 1Y7K and 6DYM in JMol Structure Visualization package along with the aligned protein sequences

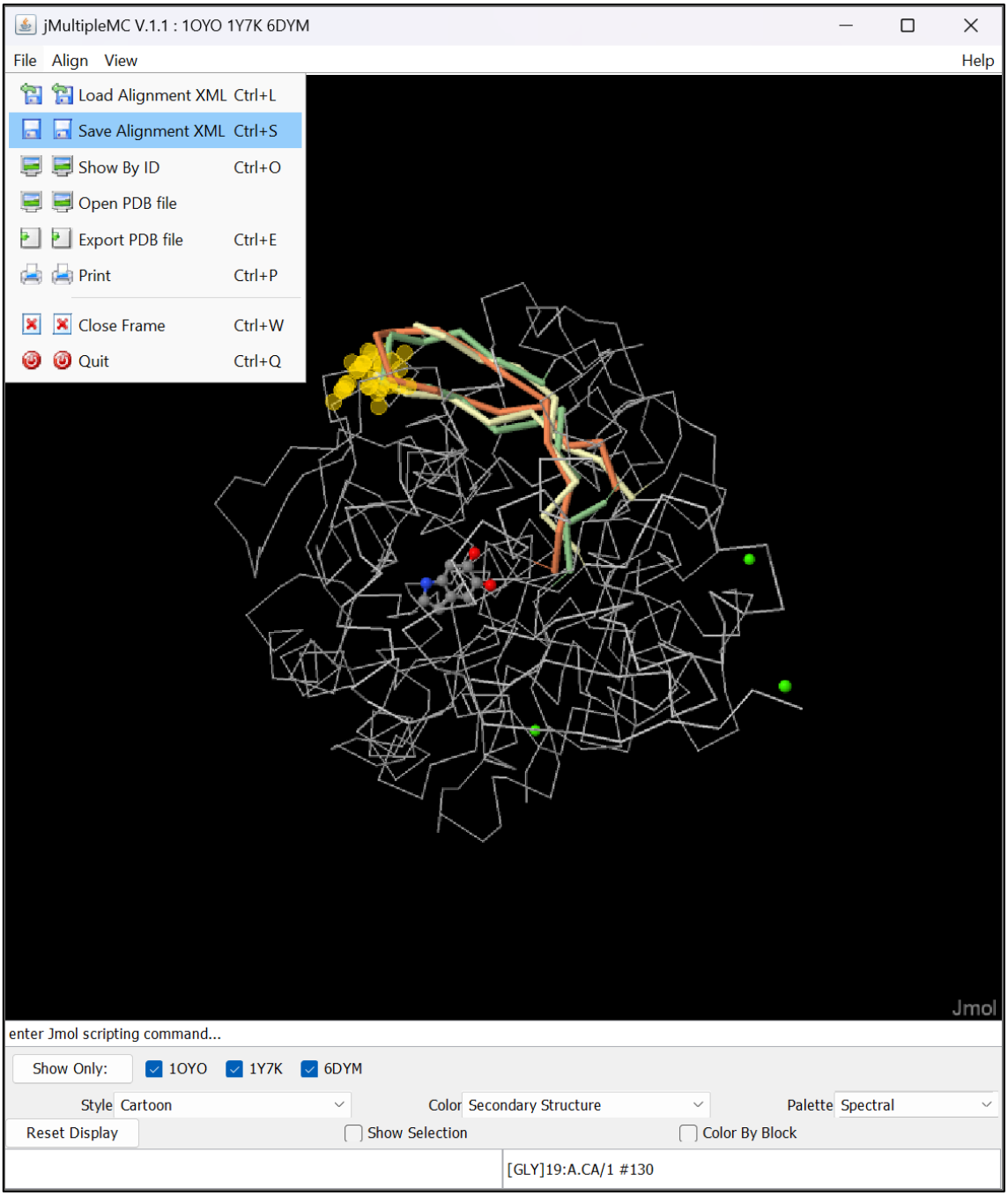


Figure 23: Option to ‘Save Alignment’ in JMol Structure Visualization package

Figure 24: Options available in the 'View' tab of JMol Structure Visualization package

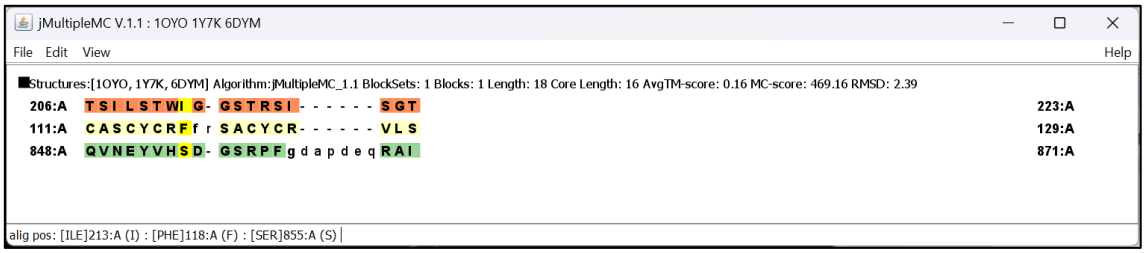


Figure 25: Result of Structural Alignment in 'Alignment Panel'

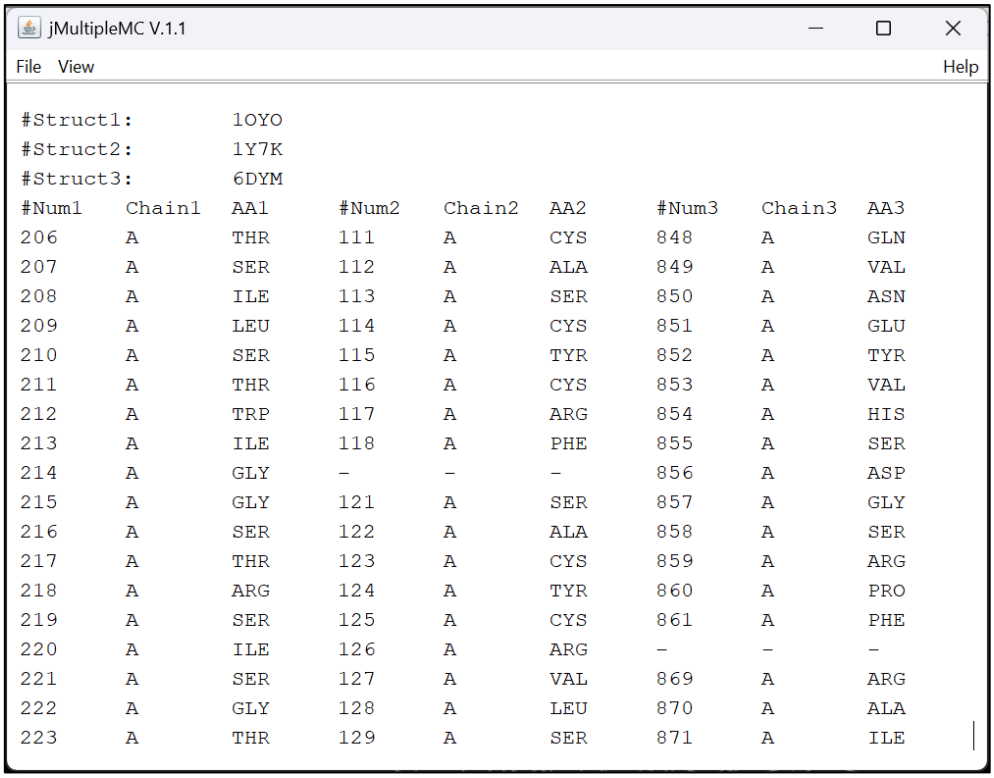


Figure 26: Result of Structural Alignment in 'View Aligned Pairs'

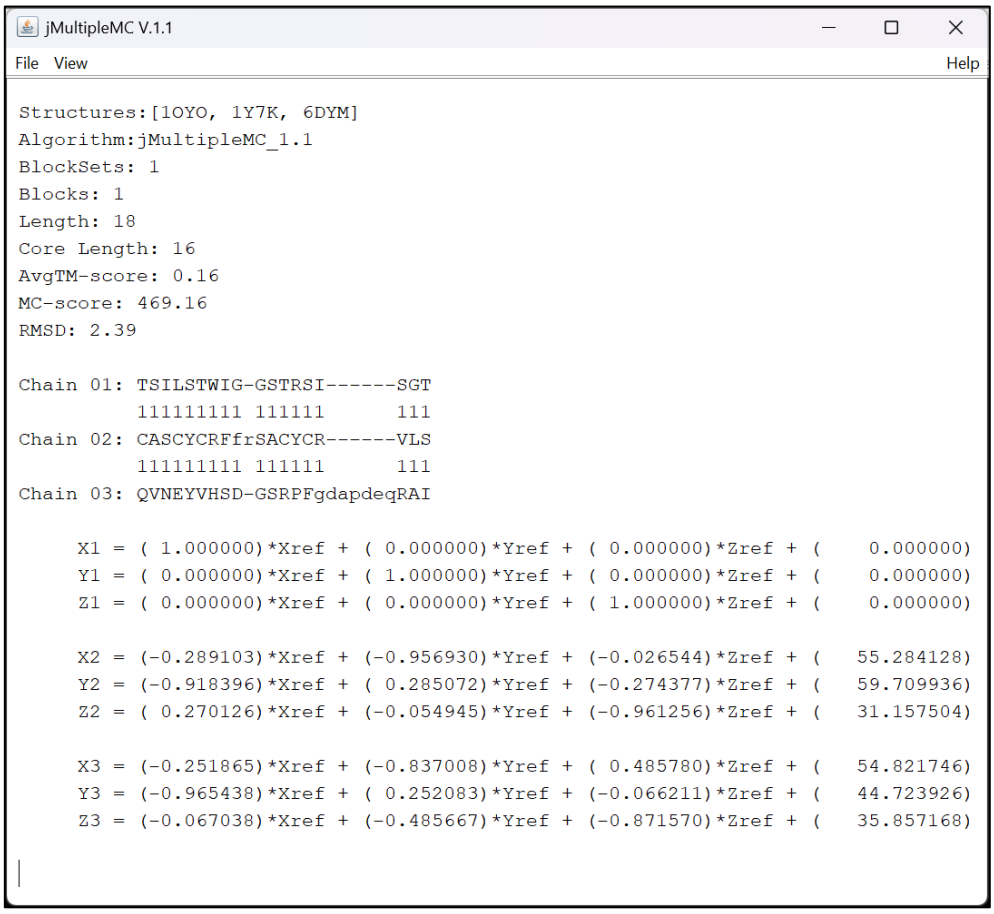


Figure 27: Result of Structural Alignment in 'View as FATCAT Result'

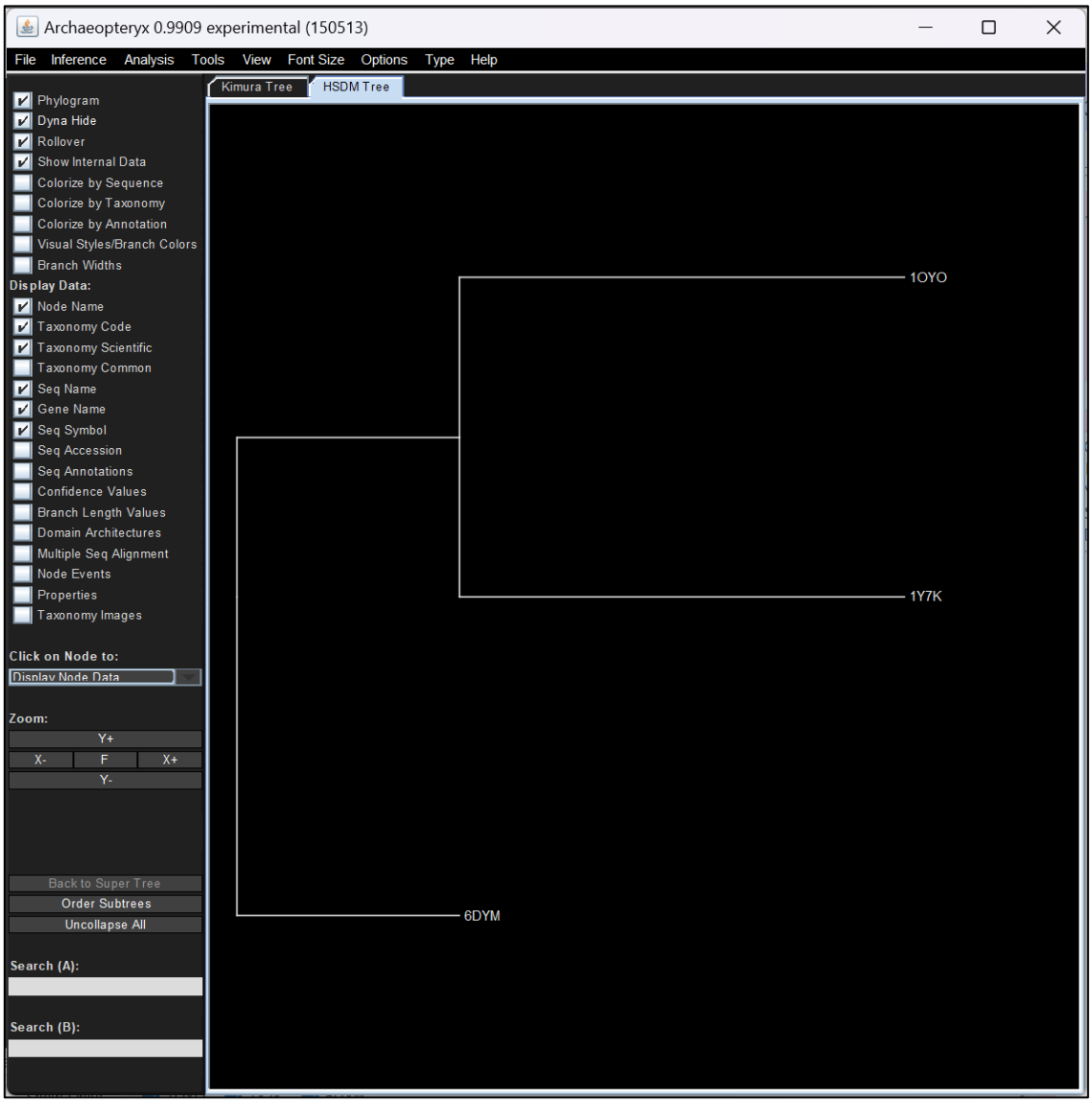


Figure 28: Result of proteins in Structural Alignment in ‘Phylogenetic Tree’

5. Literature Search

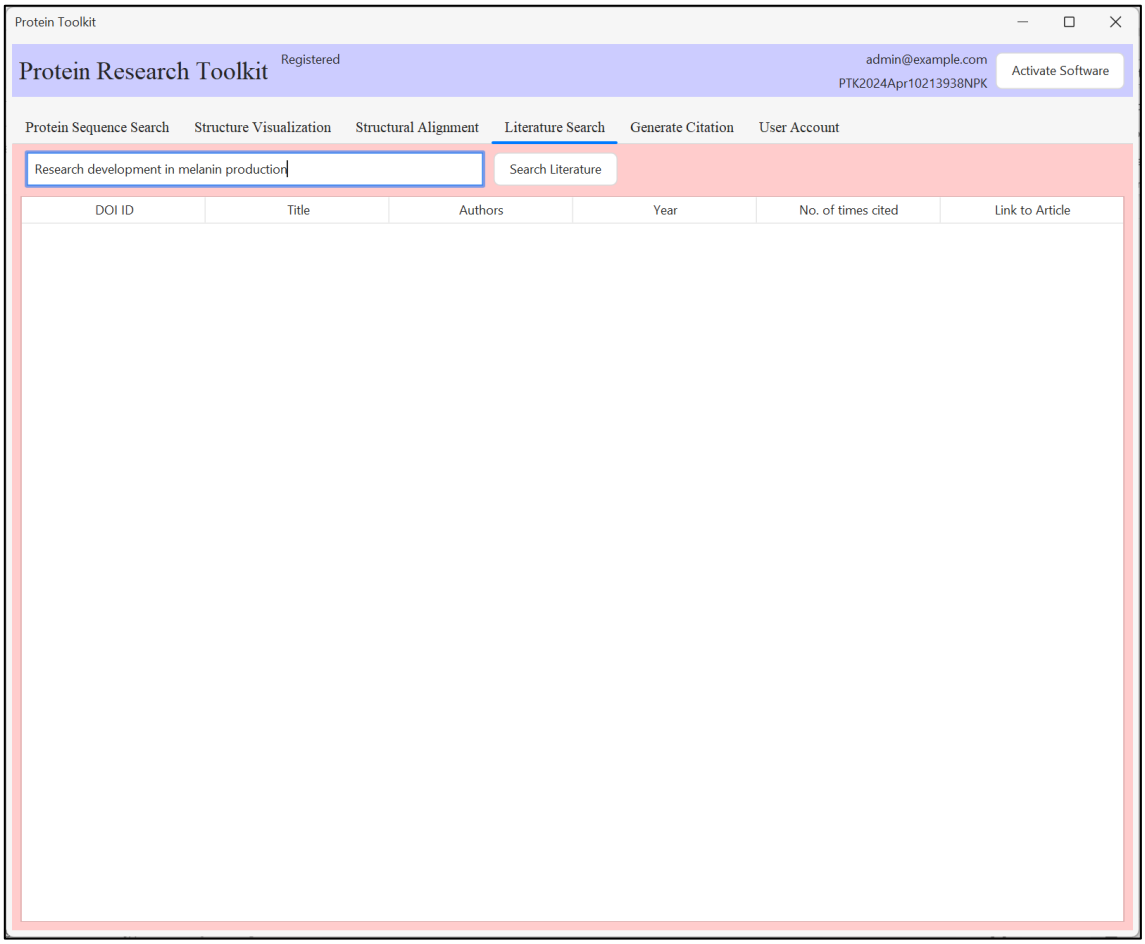
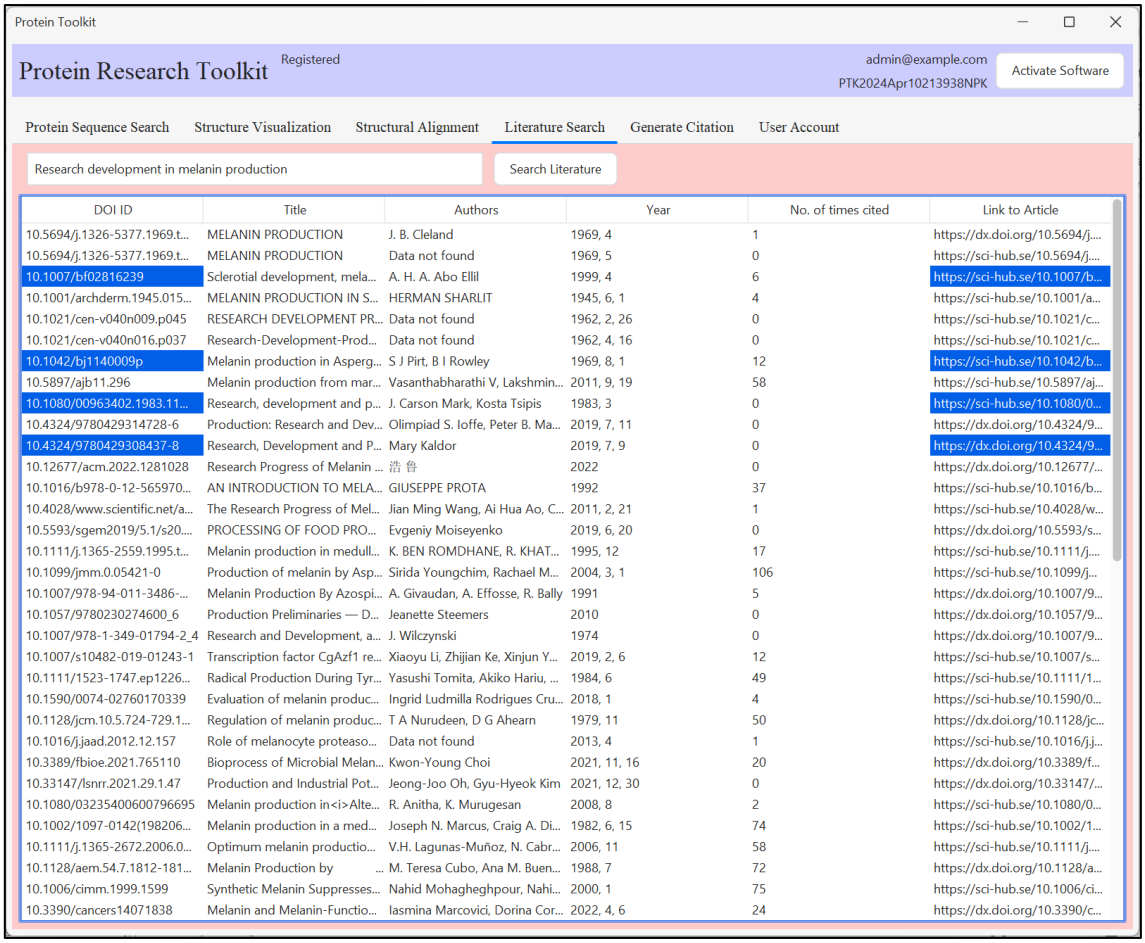


Figure 29: 'Literature Search' tool with the query 'Research development in melanin production'



DOI ID	Title	Authors	Year	No. of times cited	Link to Article
10.5694/j.1326-5377.1969.t...	MELANIN PRODUCTION	J. B. Cleland	1969, 4	1	https://dx.doi.org/10.5694/j....
10.5694/j.1326-5377.1969.t...	MELANIN PRODUCTION	Data not found	1969, 5	0	https://sci-hub.se/10.5694/j....
10.1007/bf02816239	Sclerotial development, mela...	A. H. A. Abo Ellil	1999, 4	6	https://sci-hub.se/10.1007/b...
10.1001/archderm.1945.015...	MELANIN PRODUCTION IN S...	HERMAN SHARLIT	1945, 6, 1	4	https://sci-hub.se/10.1001/a...
10.1021/cen-v040n009.p045	RESEARCH DEVELOPMENT PR...	Data not found	1962, 2, 26	0	https://sci-hub.se/10.1021/c...
10.1021/cen-v040n016.p037	Research-Development-Prod...	Data not found	1962, 4, 16	0	https://sci-hub.se/10.1021/c...
10.1042/bj1140009p	Melanin production in Asperg...	S J Pirt, B I Rowley	1969, 8, 1	12	https://sci-hub.se/10.1042/b...
10.5897/ajb11.296	Melanin production from mar...	Vasanthabharathi V, Lakshmin...	2011, 9, 19	58	https://sci-hub.se/10.5897/aj...
10.1080/00963402.1983.11...	Research, development and p...	J. Carson Mark, Kosta Tsipis	1983, 3	0	https://sci-hub.se/10.1080/0...
10.4324/9780429314728-6	Production: Research and Dev...	Olimpiad S. Ioffe, Peter B. Ma...	2019, 7, 11	0	https://dx.doi.org/10.4324/9...
10.4324/9780429308437-8	Research, Development and P...	Mary Kaldor	2019, 7, 9	0	https://dx.doi.org/10.4324/9...
10.12677/acm.2022.1281028	Research Progress of Melanin ...	浩 鲁	2022	0	https://dx.doi.org/10.12677/...
10.1016/b978-0-12-565970...	AN INTRODUCTION TO MELA...	GIUSEPPE PROTA	1992	37	https://sci-hub.se/10.1016/b...
10.4028/www.scientific.net/a...	The Research Progress of Mel...	Jian Ming Wang, Ai Hua Ao, C...	2011, 2, 21	1	https://sci-hub.se/10.4028/w...
10.5593/sgem2019/5.1/s20....	PROCESSING OF FOOD PRO...	Evgeniy Moiseyenko	2019, 6, 20	0	https://dx.doi.org/10.5593/s...
10.1111/j.1365-2559.1995.t...	Melanin production in medull...	K. BEN ROMDHANE, R. KHAT...	1995, 12	17	https://sci-hub.se/10.1111/j....
10.1099/jmm.0.05421-0	Production of melanin by Asp...	Sirida Youngchim, Rachael M...	2004, 3, 1	106	https://sci-hub.se/10.1099/j...
10.1007/978-94-011-3486-...	Melanin Production By Azospi...	A. Givaudan, A. Effosse, R. Bally	1991	5	https://dx.doi.org/10.1007/9...
10.1057/9780230274600_6	Production Preliminaries — D...	Jeanette Steemers	2010	0	https://dx.doi.org/10.1057/9...
10.1007/978-1-349-01794-2_4	Research and Development, a...	J. Wilczynski	1974	0	https://dx.doi.org/10.1007/9...
10.1007/s10482-019-01243-1	Transcription factor CgAzf1 re...	Xiaoyu Li, Zhijian Ke, Xinjun Y...	2019, 2, 6	12	https://sci-hub.se/10.1007/s...
10.1111/1523-1747.ep1226...	Radical Production During Tyr...	Yasushi Tomita, Akiko Hariu, ...	1984, 6	49	https://sci-hub.se/10.1111/1...
10.1590/0074-02760170339	Evaluation of melanin produc...	Ingrid Ludmilla Rodrigues Cru...	2018, 1	4	https://sci-hub.se/10.1590/0...
10.1128/jcm.10.5.724-729.1...	Regulation of melanin produc...	T A Nurudeen, D G Ahearn	1979, 11	50	https://dx.doi.org/10.1128/jc...
10.1016/j.jaad.2012.12.157	Role of melanocyte proteaso...	Data not found	2013, 4	1	https://sci-hub.se/10.1016/j.j...
10.3389/fbioe.2021.765110	Bioprocess of Microbial Melan...	Kwon-Young Choi	2021, 11, 16	20	https://dx.doi.org/10.3389/f...
10.33147/lsnrr.2021.29.1.47	Production and Industrial Pot...	Jeong-Joo Oh, Gyu-Hyeok Kim	2021, 12, 30	0	https://dx.doi.org/10.33147/...
10.1080/03235400600796695	Melanin production in<i>Alte...	R. Anitha, K. Murugesan	2008, 8	2	https://sci-hub.se/10.1080/0...
10.1002/1097-0142(198206...	Melanin production in a med...	Joseph N. Marcus, Craig A. Di...	1982, 6, 15	74	https://sci-hub.se/10.1002/1...
10.1111/j.1365-2672.2006.0...	Optimum melanin productio...	V.H. Lagunas-Muñoz, N. Cabr...	2006, 11	58	https://sci-hub.se/10.1111/j....
10.1128/aem.54.7.1812-181...	Melanin Production by ...	M. Teresa Cubo, Ana M. Buen...	1988, 7	72	https://dx.doi.org/10.1128/a...
10.1006/cimm.1999.1599	Synthetic Melanin Suppresses...	Nahid Mohagheghpour, Nahi...	2000, 1	75	https://sci-hub.se/10.1006/ci...
10.3390/cancers14071838	Melanin and Melanin-Functio...	Iasmina Marcovici, Dorina Cor...	2022, 4, 6	24	https://dx.doi.org/10.3390/c...

Figure 30: Results obtained in the ‘Literature Search’ tool for the query ‘Research development in melanin production’

6. Generate Citation

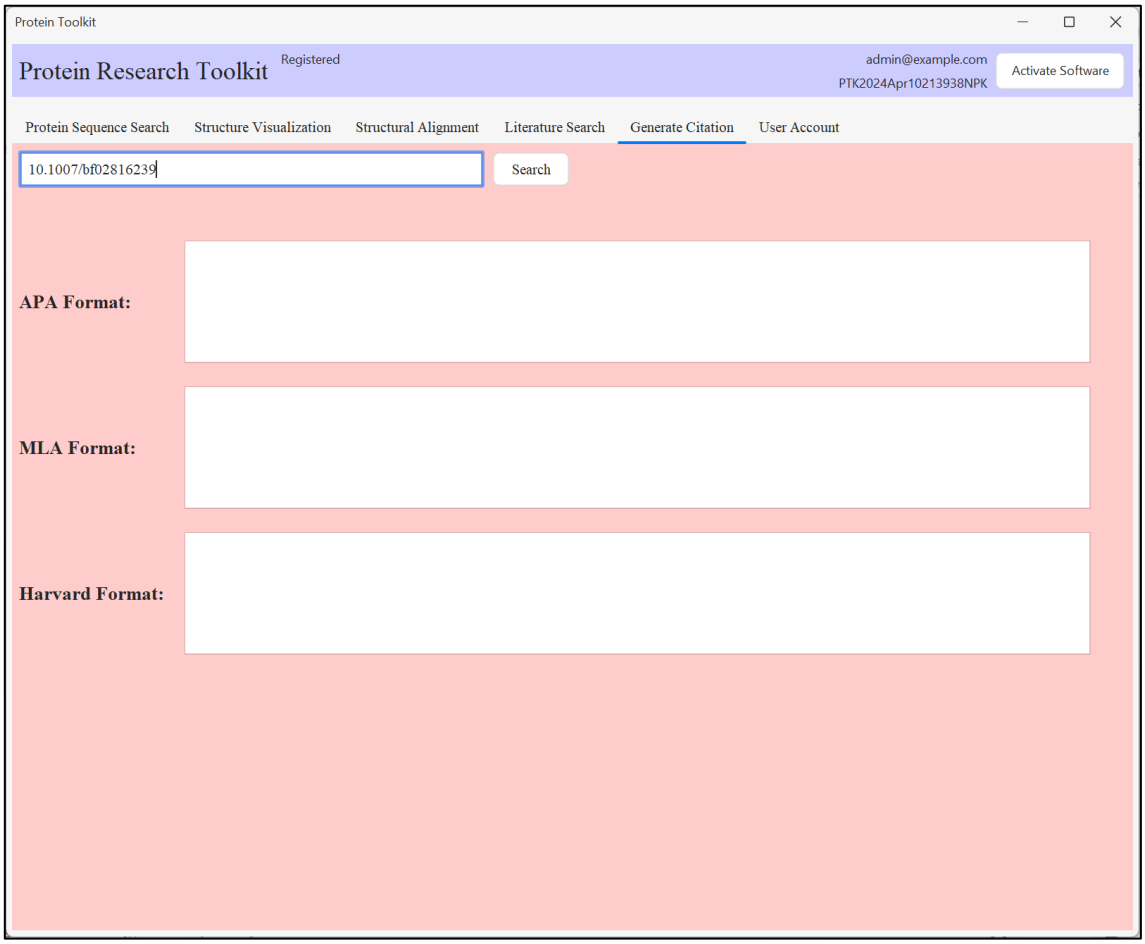


Figure 31: ‘Generate Citation’ tool with the query DOI ID: ‘10.1007/bf02816239’

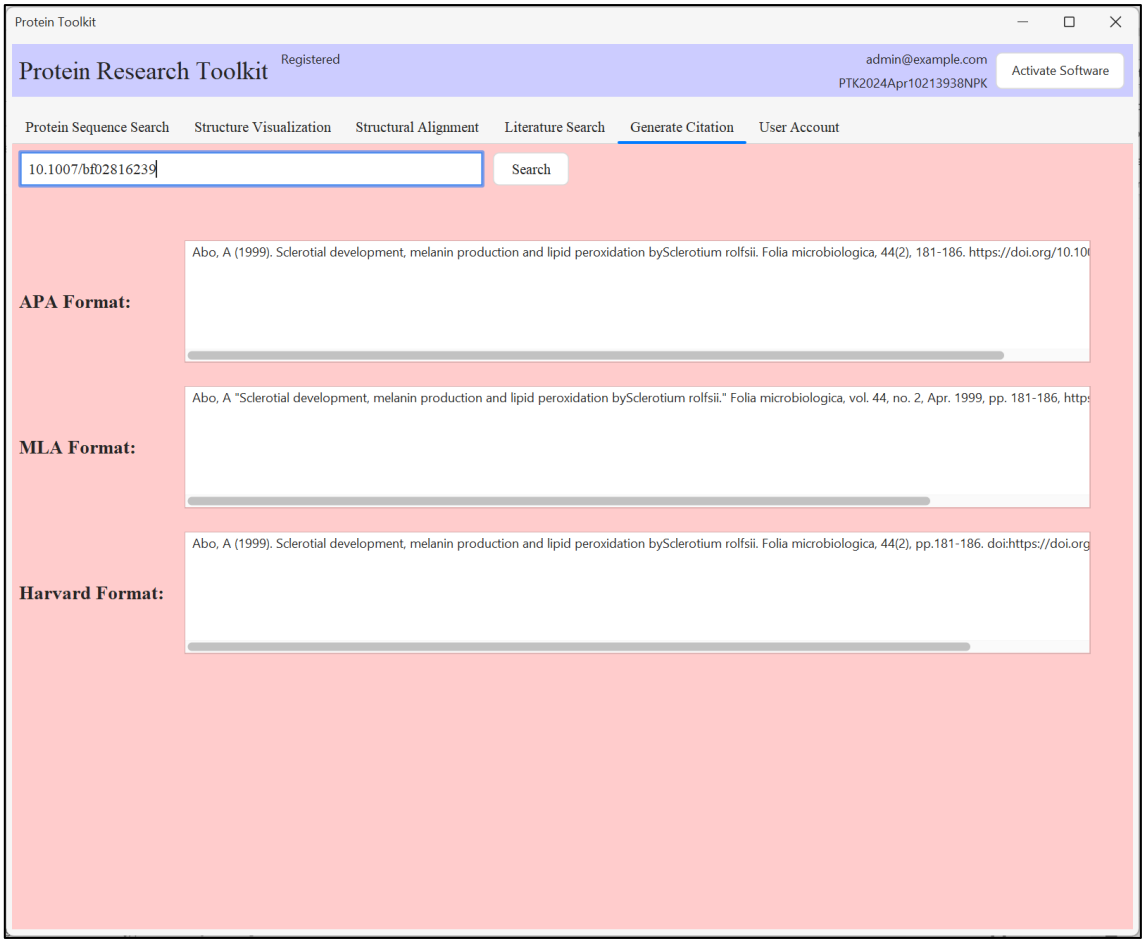


Figure 32: Results obtained for the DOI ID: 10.1007/bf02816239 in the following 3 formats: APA (7th ed.) Format, MLA (9th ed.) Format and Harvard (International) Format

RESULTS:

1. The database has been created on an online server and has been accessed using JDBC's MySQL driver. The buttons 'Register' and 'Login' have been programmed to execute certain queries on the MySQL server to store and retrieve data respectively. The retrieved data is then stored at a secured location in the local system.
2. During registration, an automatic license key is generated which is then stored onto the MySQL server and is retrieved as well as stored at a secured location in the local system, during the login process.
3. The UniProt Search Engine has been reverse engineered to obtain its URL endpoint and is further used in the 'Protein Sequence Search' tool for retrieving data directly from the UniProt database. The response received was then converted into the tabular format.
4. The Protein Data Bank (PDB) Search's REST-API endpoint has been used to retrieve the PDB IDs from the PDB database. The retrieved PDB IDs have been queried to the PDB data's REST-API endpoint. The response received was then converted into the tabular format.
5. BioJava package has been used to retrieve the protein structures from the PDB database. The protein structures are then visualized using JMol Structure Visualization package.
6. BioJava package has been used to retrieve the protein structures from the PDB database and the structures have been aligned using FATCAT (flexible) algorithm. The aligned protein structures are then visualized using JMol Structure Visualization package.
7. The Crossref's REST-API has been used to perform the literature search and retrieve relevant information like DOI ID, title, authors, year of publication and citation count. The research papers are searched in SciHub database as well as DOI database and those links have also been integrated into the table along with the data retrieved from the Crossref.
8. The mybib.com's citation machine has been reverse engineered to obtain the REST-API endpoint which is then integrated into the Citation Generator tool to generate citations in APA format, MLA format and Harvard format using DOI ID.

CONCLUSION:

In the ever-evolving field of bioinformatics, researchers require efficient and user-centric tools to delve deeper into the complexities of proteins. By offering a comprehensive suite of protein analysis tools, all accessible from a user-friendly platform, the Protein Research Toolkit will significantly streamline research workflows and empower scientists to make innovative discoveries in the field of protein science.