

**COMPUTER SCIENCE DEPARTMENT**

**DOCUMENTATION ON JAVA-SWING BASED VOTING SYSTEM (GROUP-7)**

**Background:**

Voting systems are systems designs to help conduct a free and fair election. Since the introduction of these system types, it has helped to a drastic decrease in misunderstanding and confusions that evolve from voting or selection of leaders through conduction elections or voting making this system nearly ubiquitous.

Basically, voting systems are developed with more than three different interfaces or pages which each of these serve as a subsystem for the main voting system and are integrated to form a complete voting system. Some of these subsystems include a login or User authentication (subsystem to ensure security for the system), voting page (where the authenticated user is allowed to select his / her vote choices), a result or a poll page (A subsystem to calculate and display the total votes casted to a particular candidate and number of people who had already voted, and many others. These subsystems are developed and integrated as a result of problems aroused while using the manual approach or an old voting system.

**Problem Statement:**

During the development process we encountered many problems some of which include:

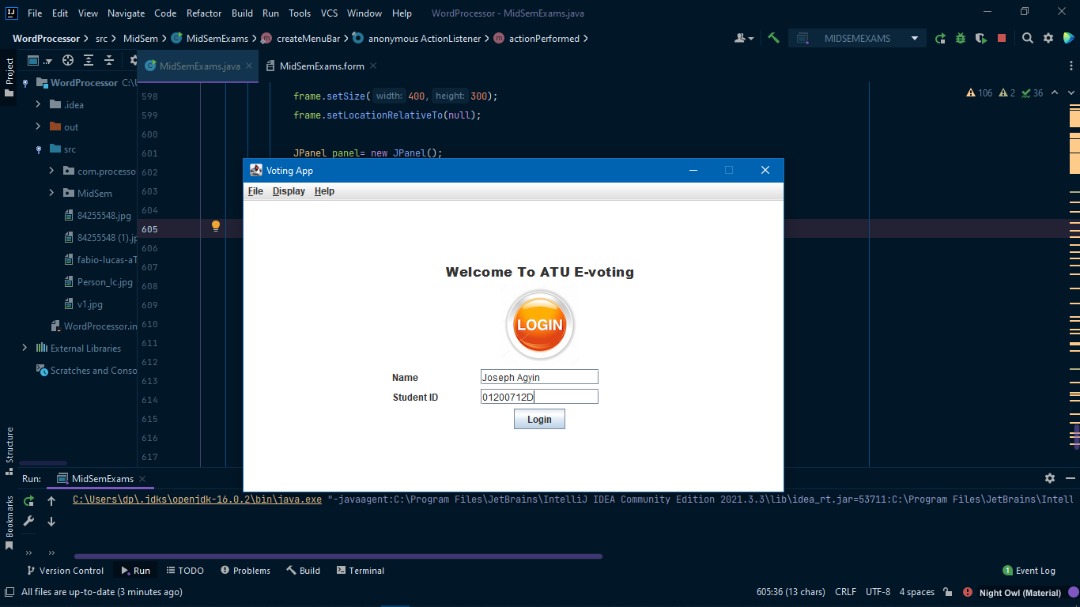
* **Difficulty in Integrating the poll subsystem into the main voting system**, in the sense that Accumulation of votes, casted by a user’s from the voting page could not reflect to the poll or the results page as expected. we created an action listener for all vote check boxes and the continue Buttons so that when clicked at the same time the individual text fields in the result menu item should, Display that count but unfortunately, it wasn't increasing as we expected, but rather display the initial number we declared. So, we had to figure out a way to make the possible.

**Aims and Objectives:**

The Aims and Objectives for creating this system is to:

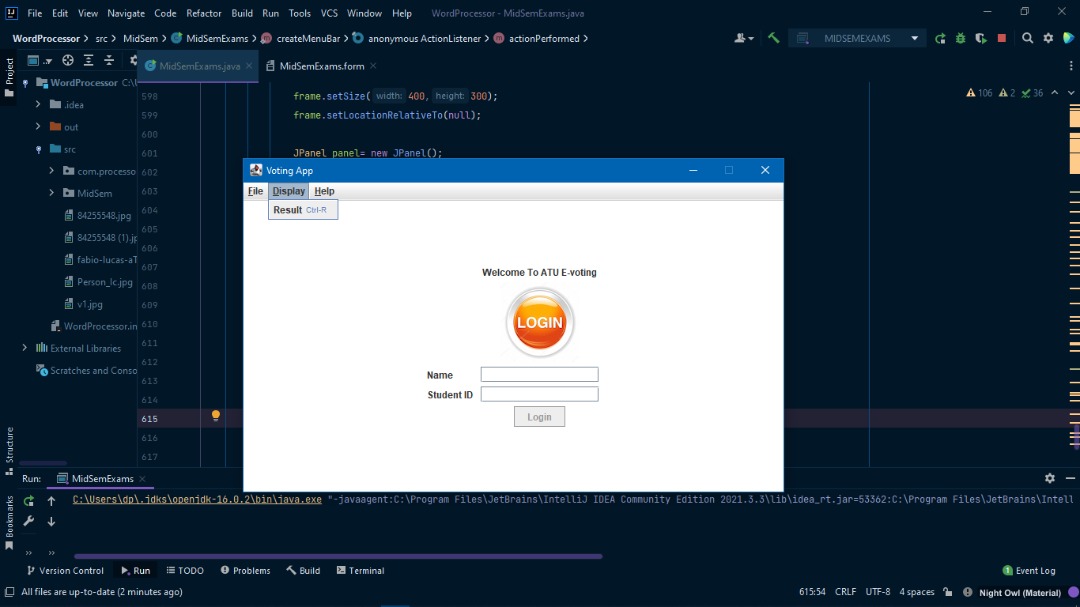
1. Ensure accuracy in counting votes casted without any difficulties.
2. Reduce overvoting and rejection of votes that used to occur during manual voting.
3. Creating a robust method in managing votes casted for each candidate.
4. Lastly allow system administrators to add candidates with ease.
5. To maximize student participation in voting (since it ill be faster process).

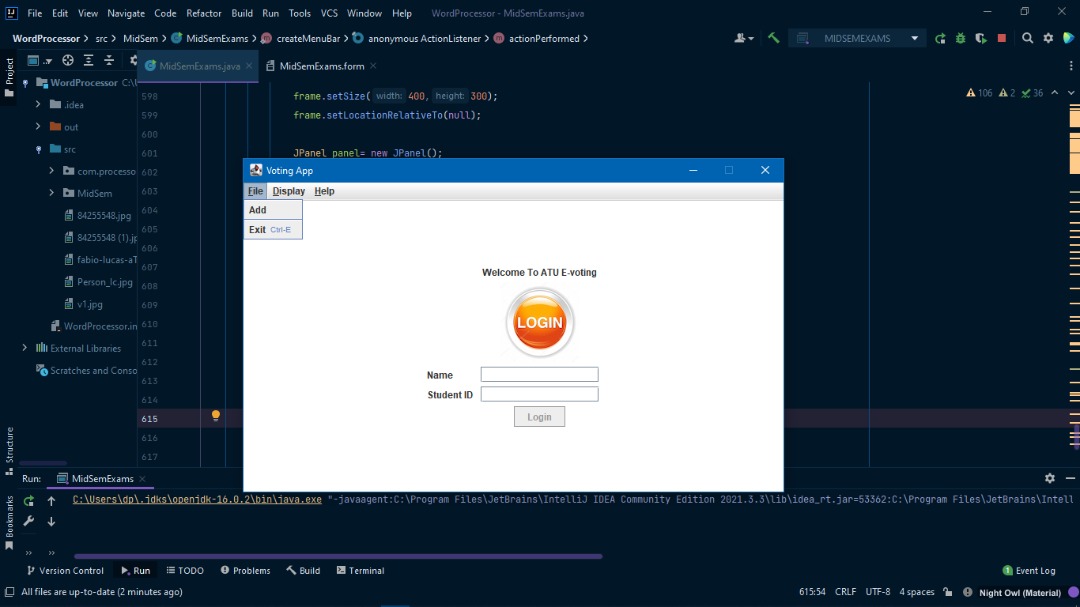
GRAPHICAL USER INTERFACE

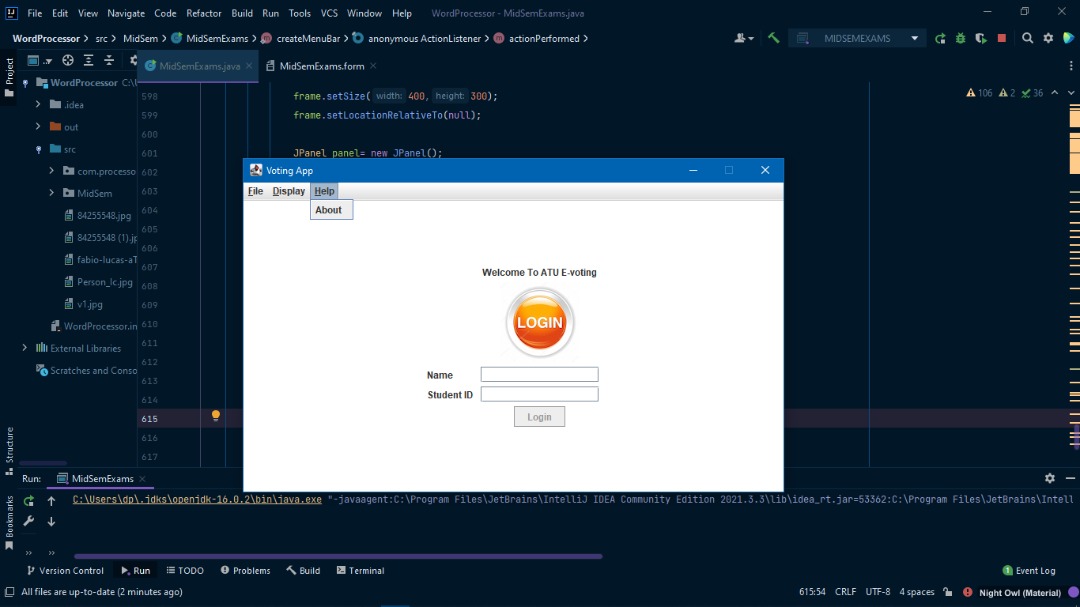
****

**Login Page:**

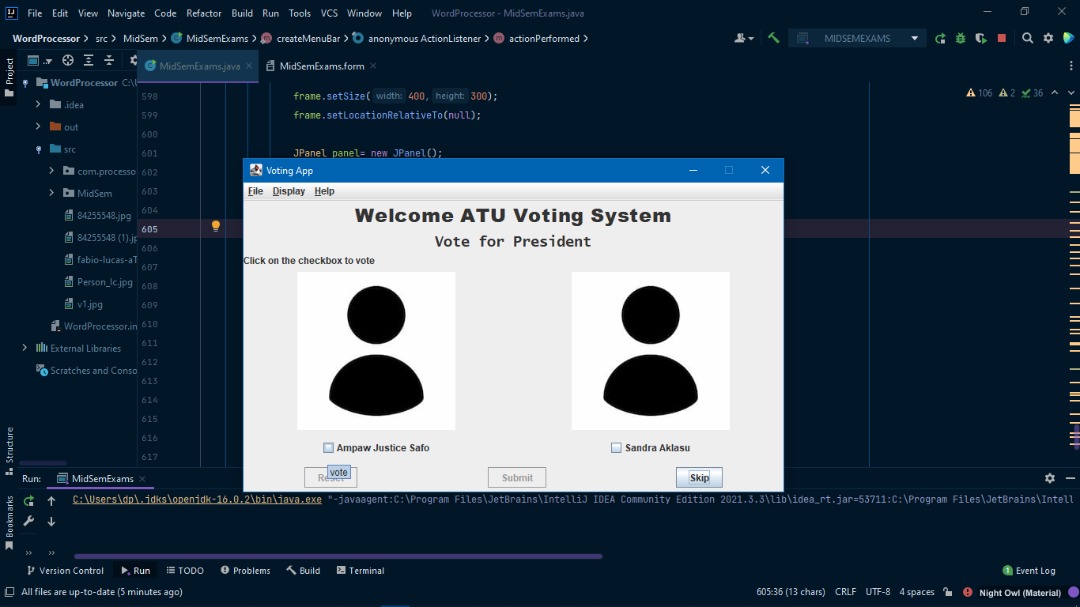
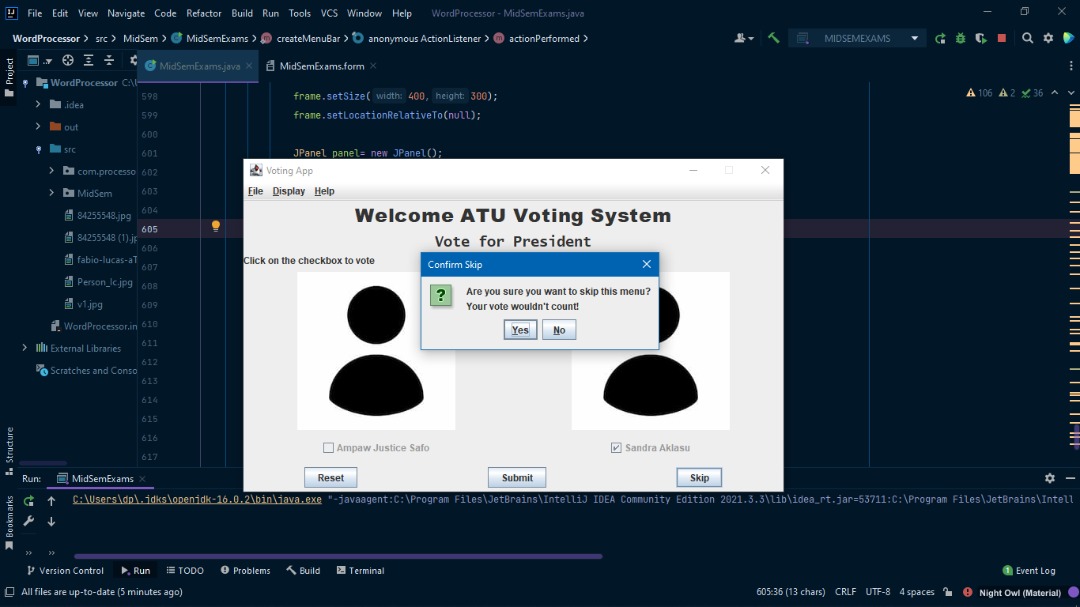
Where user credentials such as Name and student Id, is being collected for Authentications purposes.

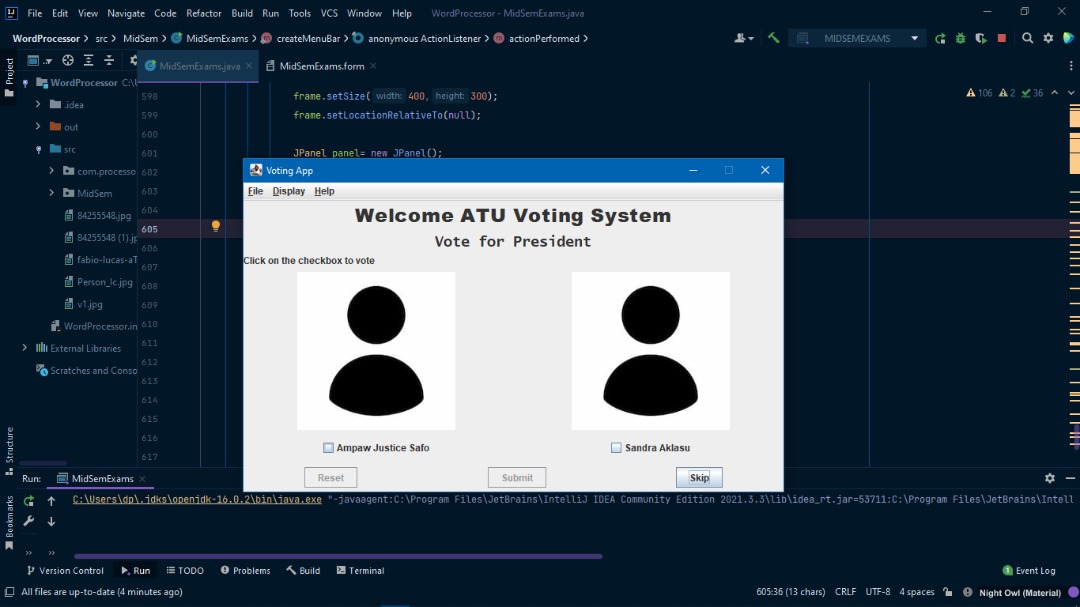
****

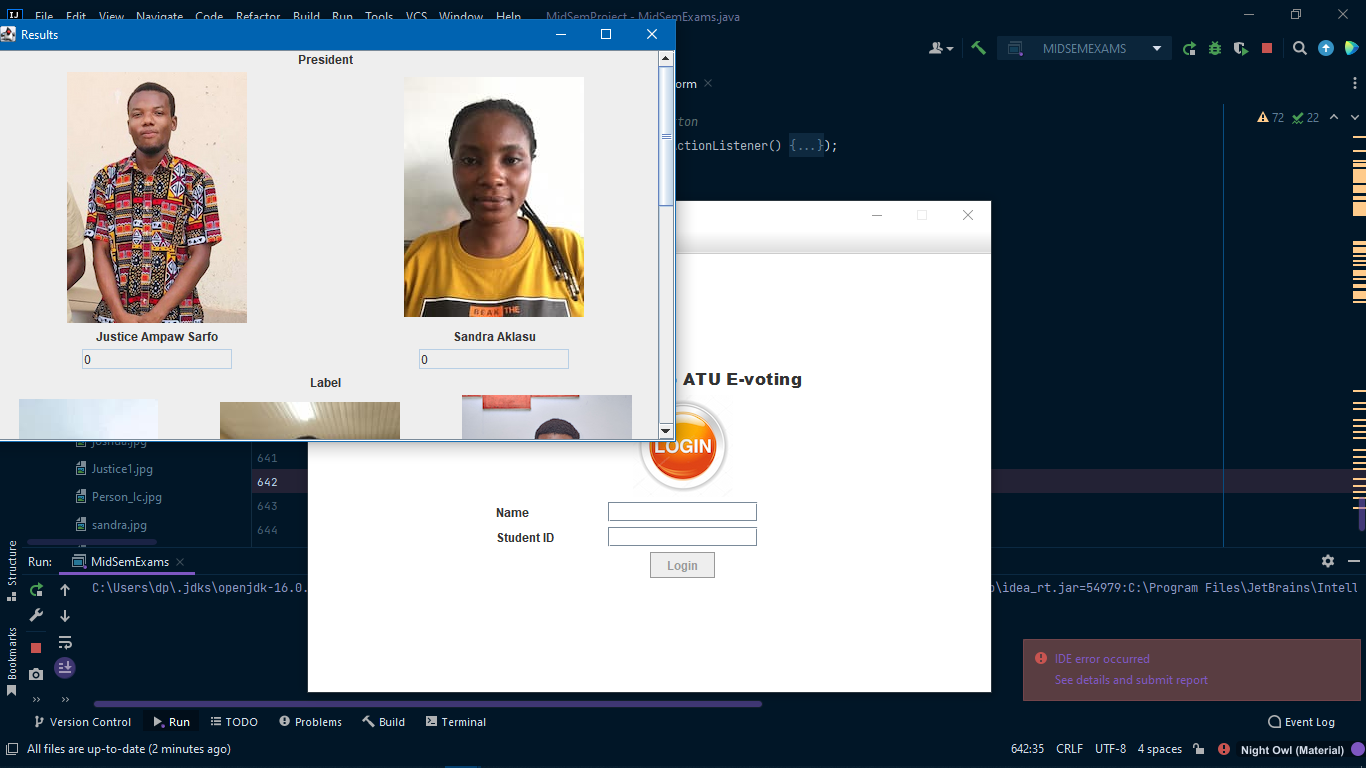
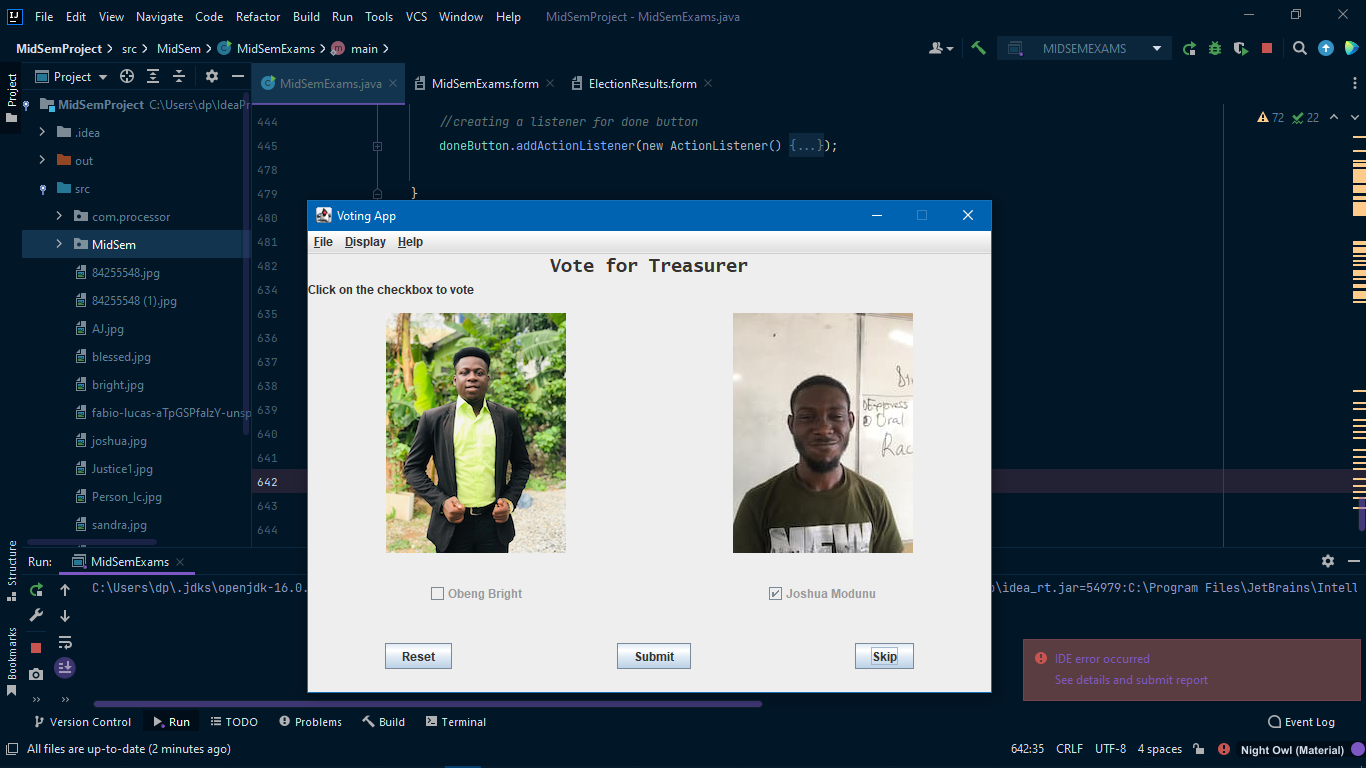
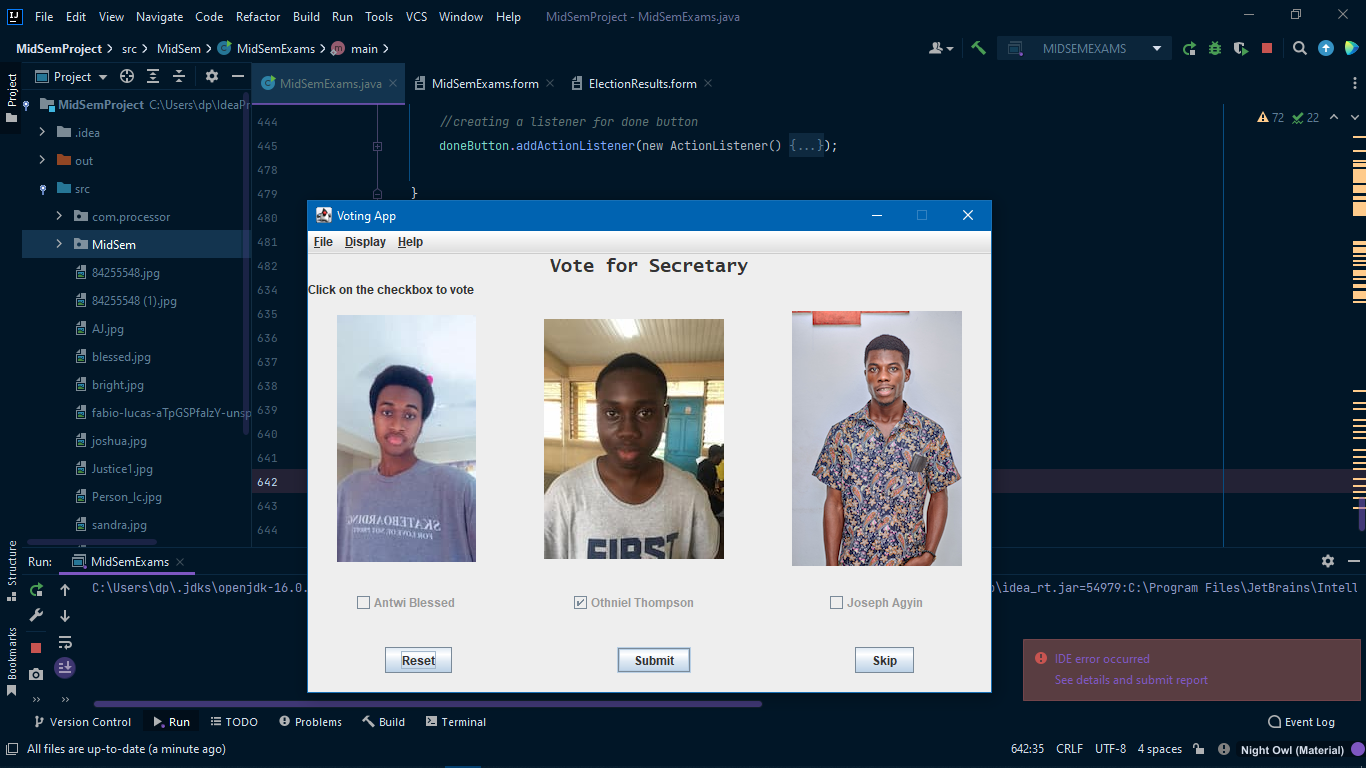
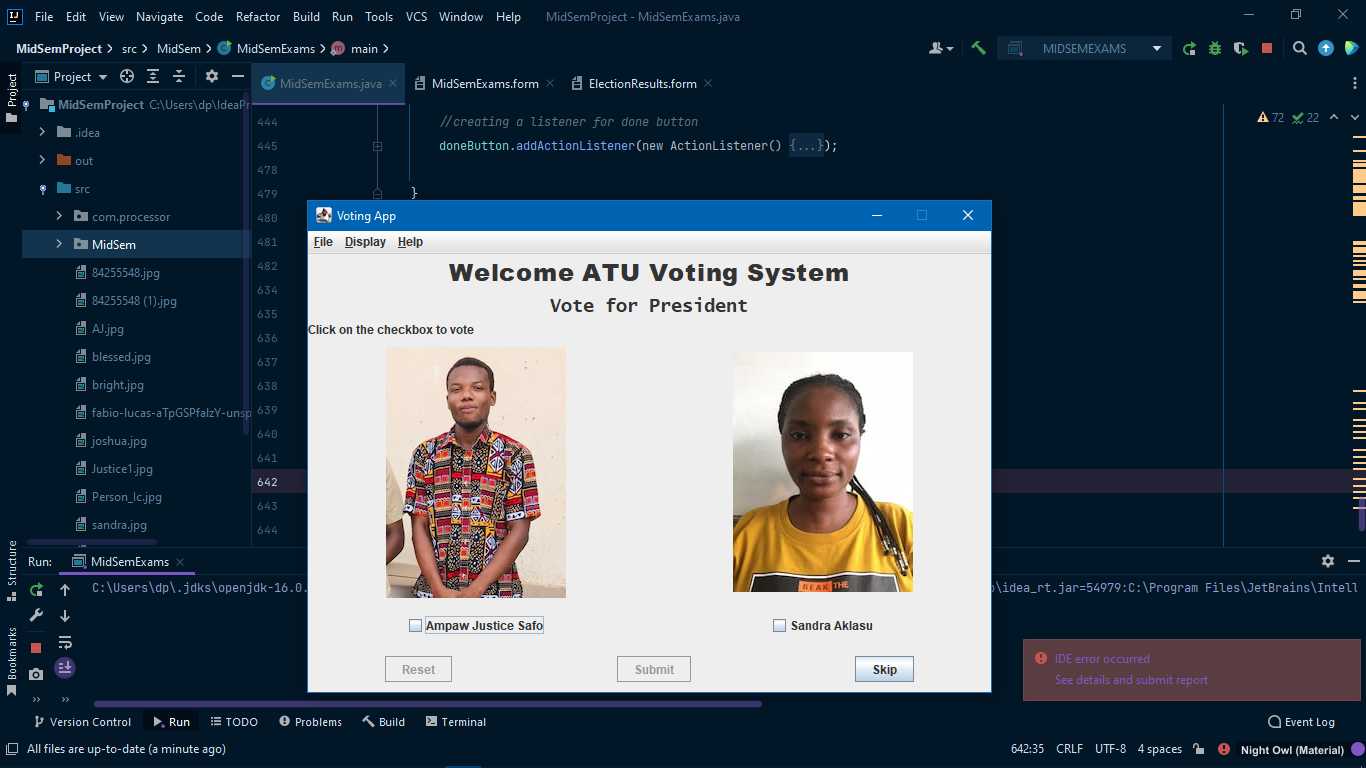
****

****

**Voting Page:**

****

****

****

**Swing Components :**

* JPanel
* JLabel
* JButton
* JCheckBox
* JScrollPane
* JTextField
* JComboBox

**Event handled:**

* javax.swing.\*;
* java.awt.\*;
* java.awt.event.\*;
* java.util.ArrayList;

ASSESSMENT FORM

[To be printed and attached to Documentation]

CLASS (A, B, C or PT): \_2B\_\_\_\_\_ GROUP: \_\_\_GROUP 7

# GROUP MARKS [40 Marks]

|  |  |  |
| --- | --- | --- |
| BUILD APPROACH | MARKS |  |
| GUI using Swing (Names & Pictures) |  | /7 |
| Functionality (Accumulation of Votes) |  | /5 |
| Menu bar (with Mnemonics and Accelerators) |  | /3 |
| Display of Results |  | /5 |
| DOCUMENTATION |  |  |
| Background |  | /4 |
| Problem Statement |  | /4 |
| Objective |  | /2 |
| Design and Implementation |  | /10 |
| TOTAL |  | /40 |

# MEMBERS CONTRIBUTION [10 Marks]

|  |  |  |  |
| --- | --- | --- | --- |
| S/N | MEMBERS | INDEX NO. | MARKS |
| 1 | Ampaw Justice Safo | 01203769D | /10 |
| 2 | Joshua Modunu | 01204961D | /10 |
| 3 | Joseph Agyin | 01200712D | /10 |
| 4 | Blessed Antwi | 01200329D | /10 |
| 5 | Obeng Bright | 01200663D | /10 |
| 6 | Sandra Aklasu | 01203745D | /10 |
| 7 | James Mckeon Awuah | 01202207D | /10 |
| 8 | Thompson Othniel | 01203721D | /10 |
| 9 | Gankui Wisdom Kwame | 01200035D | /10 |
| 10 | Umme Khadija Mustapha- Sey | 01201185D | /10 |
| 11 | Bernard Asiamah Ankah | 01202159D | /10 |