**Report: Light Switch Application**

**1. Introduction**

The Light Switch Application is a simple desktop-based program developed in Java using the Swing library. The application simulates the functionality of a light switch and visually represents the bulb's ON/OFF states using images. It incorporates features like manual toggling, automatic toggling based on time, and an intuitive user interface.

**2. Features Implemented**

**2.1 Manual Light Toggle**

The application allows users to manually toggle the light on and off using a button. The button's label changes dynamically to reflect the current state (“Turn ON” or “Turn OFF”).

**2.2 Automatic Light Toggle**

The application includes an automatic toggle feature based on the time of day:

* **Night Time (6:00 PM to 6:00 AM):** The light automatically turns on.
* **Day Time (6:00 AM to 6:00 PM):** The light automatically turns off.

This feature is implemented using Java’s Timer and Timer Task classes, which check the system time every minute to update the light’s state.

**2.3 Visual Representation of the Bulb**

The application uses two images (“light\_on.jpeg” and “light\_off.jpeg”) to represent the bulb’s state. The images are resized dynamically to fit the user interface.

**2.4 Dynamic Background Color**

The panel background changes color depending on the light’s state:

* **ON:** Background is white.
* **OFF:** Background is black.

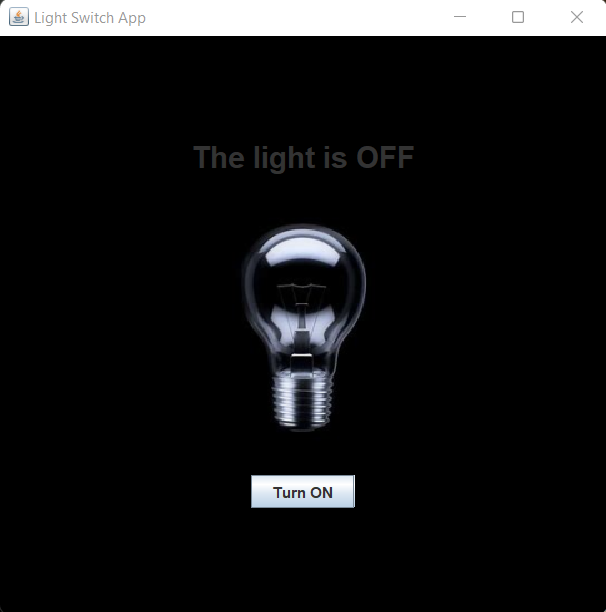
**2.5 Intuitive User Interface**

* The layout uses GridBagLayout for component alignment.
* Labels, buttons, and images are organized for clarity and simplicity.

**3. Screenshots of the Project Interface**

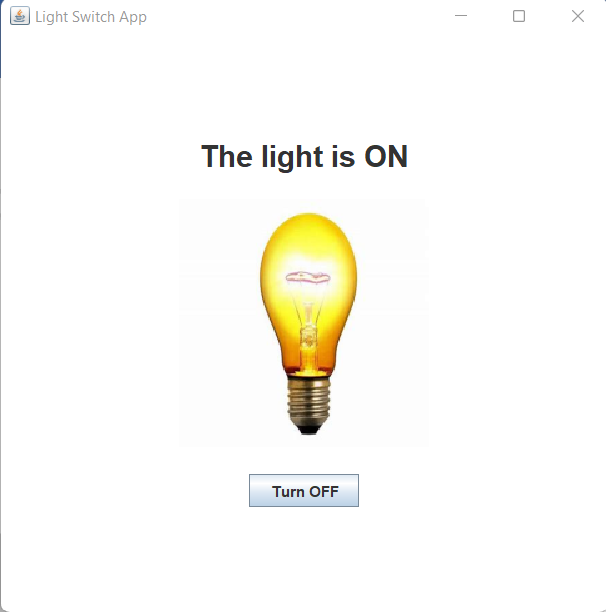
**3.1 Initial State**

When the application starts, the light is OFF. The button displays “Turn ON,” and the screen background is black.



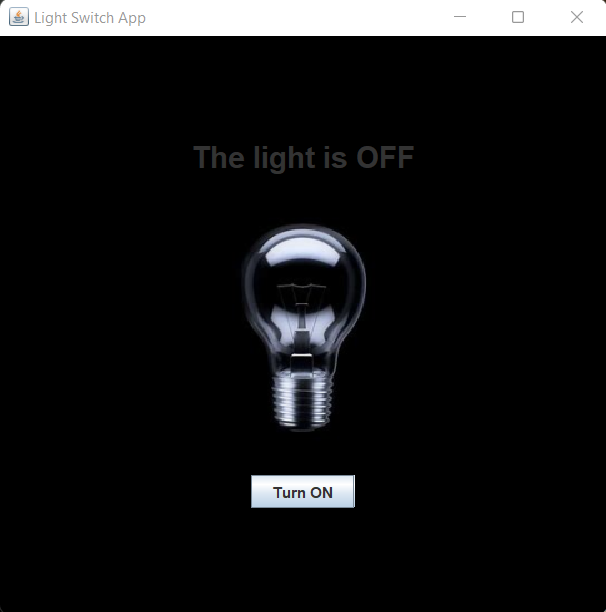
**3.2 Light ON**

When the button is clicked or during automatic toggling (at night), the light turns ON. The button changes to “Turn OFF,” and the bulb image updates to reflect the ON state.



**3.3 Light OFF**

When toggled manually or automatically (during the day), the light turns OFF. The background reverts to black, and the bulb image updates accordingly.



**4. Challenges Faced During Development**

**4.1 Image Resizing and Placement**

Integrating images into the JLabel component and resizing them dynamically to maintain aspect ratio was initially challenging. This was resolved using the getScaledInstance() method of the Image class.

**4.2 Time-Based Automatic Toggle**

Implementing a timer to monitor the system’s local time and toggle the light appropriately required careful handling of concurrency. Using a separate thread with Java’s Timer class helped address this.

**4.3 Dynamic Layout Management**

Aligning components (label, image, and button) in the panel to create an intuitive layout posed challenges. GridBagLayout provided the flexibility needed for precise positioning.

**4.4 Testing and Debugging**

Ensuring that the automatic toggling worked seamlessly under different time conditions required extensive testing by manually changing the system clock.

**5. Conclusion**

The Light Switch Application successfully simulates a real-world light switch with both manual and automatic functionality. The application’s simplicity and visual feedback make it easy to use. The project demonstrated the effectiveness of Java Swing in developing graphical user interfaces and highlighted practical solutions to challenges such as layout management and time-based operations.