

JavaNote - Text-Based Note Taking Application

Mudunoori Rohan Raj

20BRS1207

ABSTRACT

JavaNote is a simple text-based note-taking application designed to help users create, organize, and manage their notes efficiently. With features like formatting, categorization, and basic search functionality, JavaNote provides a user-friendly interface for maintaining digital notes. This project is an excellent opportunity to practice Java programming and GUI development skills.

KEY FEATURES

1. User Authentication: Users can create accounts or login securely to access their notes.
2. Note Creation and Editing: Users can create new notes, edit existing ones, and apply basic formatting (such as bold, italic, and underline) using a user-friendly text editor.
3. Categorization: Notes can be categorized into different topics or folders, allowing users to better organize their content.

4. Search Functionality: Users can search for specific notes by keywords, titles, or tags.

1. Search Functionality: Users can search for specific notes by keywords, titles, or tags

2. User Interface: A graphical user interface (GUI) will be provided for users to interact with the application. The GUI should include buttons, menus, and text areas for creating, editing, and managing notes.

3. File Persistence: Notes and user data will be saved locally using file storage. Serialization can be used to save and load notes.

4. Formatting Options: Implement a set of buttons or formatting shortcuts that users can use to apply formatting options to selected text in the note editor.

5. Responsive Design: Design the application's GUI to be responsive and user-friendly, ensuring that it works well on different screen sizes.

6. Data Validation: Implement input validation to ensure that users enter valid data when creating or editing notes

Project Structure

Create a directory structure like this for your project:

```
TextNoteTakingApp/  
├── src/  
│   ├── Main.java  
│   └── Note.java  
├── lib/  
├── resources/  
│   ├── application.css  
│   └── Note.fxml  
└── out/
```

Dependencies

we'll need to download and add the JavaFX libraries to your project. This can be done by configuring the build system (e.g., using Gradle or Maven) or by downloading the JavaFX SDK from the official website and including it in your project.

Main.java

```
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) {
        try {
            FXMLLoader loader = new FXMLLoader(getClass().getResource("/Note.fxml"));
            BorderPane root = (BorderPane) loader.load();
            Scene scene = new Scene(root, 600, 400);

            scene.getStylesheets().add(getClass().getResource("/application.css").toExternalForm());

            primaryStage.setScene(scene);
            primaryStage.setTitle("JavaNote - Text-Based Note Taking App");
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Note.java

```

import javafx.fxml.FXML;
import javafx.scene.control.TextArea;

public class NoteController {

    @FXML
    private TextArea noteTextArea;

    public void saveNote() {
        String noteText = noteTextArea.getText();
        // Implement the code to save the note text to a file or database.
    }

    public void loadNote() {
        // Implement the code to load a previously saved note.
    }
}
...

```

Note.fxml

This is the FXML file for the user interface of your application. You can design the layout as you prefer.

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.Button?>
<?import javafx.geometry.Insets?>

<BorderPane xmlns:fx="http://javafx.com/fxml" fx:controller="NoteController">
    <center>
        <TextArea fx:id="noteTextArea" promptText="Type your notes here..."
BorderPane.alignment="CENTER">
            <padding>
                <Insets top="10" right="10" bottom="10" left="10"/>
            </padding>
        </TextArea>
    </center>
    <bottom>

```

```
<Button text="Save" onAction="#saveNote" BorderPane.alignment="CENTER"/>
<Button text="Load" onAction="#loadNote" BorderPane.alignment="CENTER"/>
</bottom>
</BorderPane>
...

```

application.css

We can create a CSS file to style your application.

Building and Running

To build and run a JavaFX application, we have to set up the JavaFX environment properly and then compile and run the `Main.java` class.