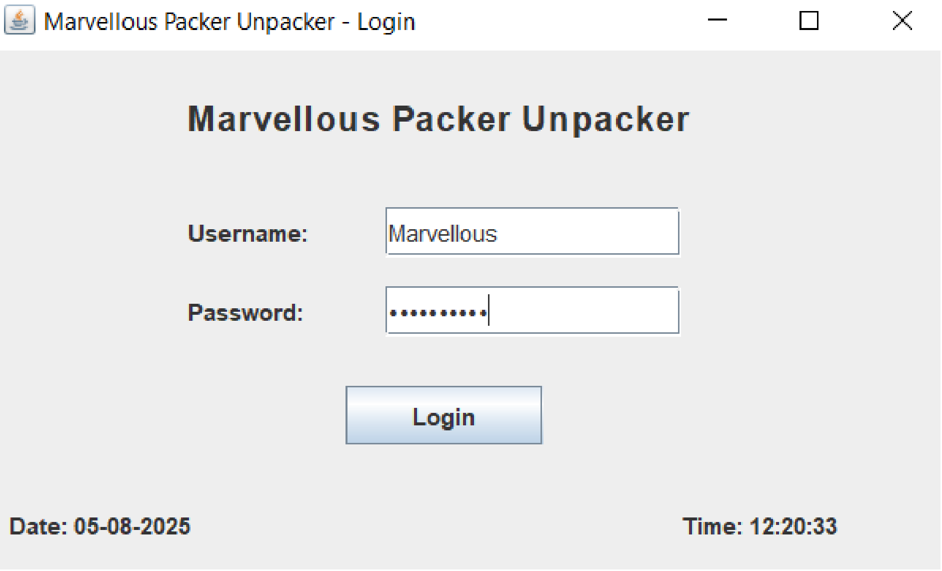
File Packer Unpacker Utility

This project is a robust, platform-independent utility designed to securely manage files through consolidation and extraction. It achieves this by leveraging Java for both its graphical user interface (GUI) and its underlying processing logic.

- Packing Activity: The utility takes multiple files from a specified directory and merges them into a single packed file. This process includes creating encrypted metadata (header) and writing the encrypted data sequentially. Security is ensured by using a user-defined encryption key.
- Unpacking Activity: The system reads the packed file, uses the matching decryption key to decrypt the metadata and data, and accurately recreates all original files based on their size and name.
- Platform Independence: By being entirely written in Java, the application guarantees that the packing and unpacking functions operate consistently across various operating systems.

Steps to develop above project

- Design one window which accept username and password from user for authentication purpose.



Marvellous Packer Unpacker

Username: Marvellous

Password: ••••••••••

Date: 05-08-2025

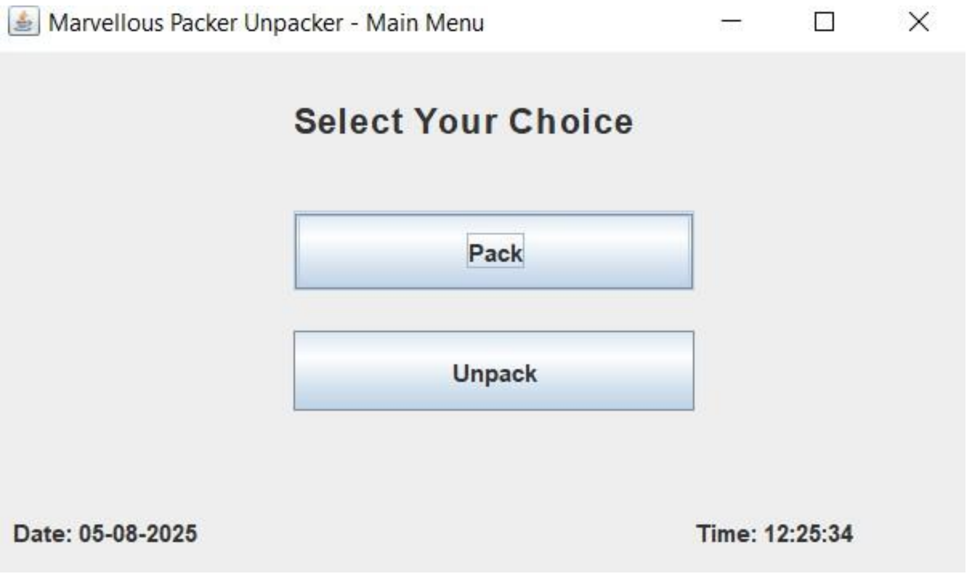
Time: 12:20:33

- When user submit the username and password we have to check whether is
- Username: Marvellous
- Password: Marvellous.

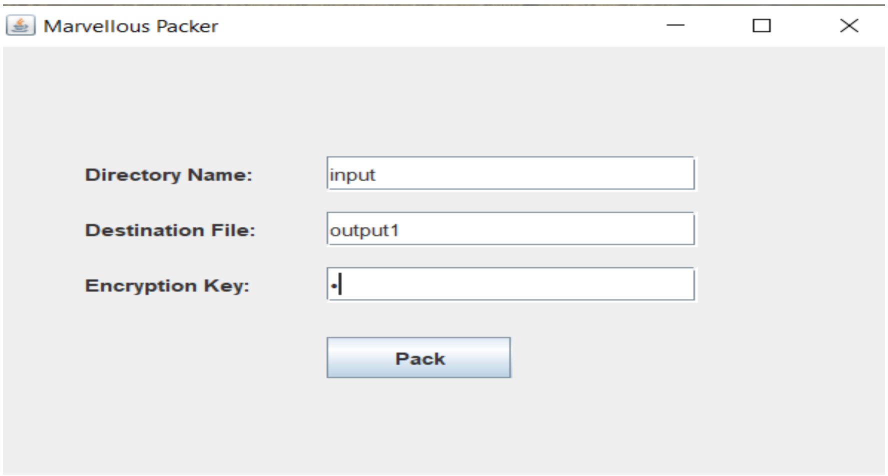
- If password or username is incorrect then we have to provide 3 attempts to user. I user is unable to provide correct username and password in 3 attempts we have to close the project.

- For checking whether username and password contains minimum contains 8 letters or not we have to create one thread. That thread also check whether caps lock key is on or not.

- After successful authentication we have to open new window which displays two options as Pack and Unpack.



- When user press Pack button we have to open new window which looks like

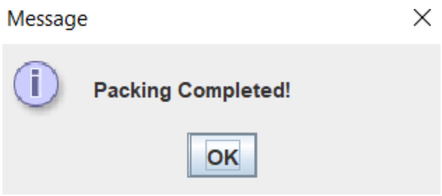


Directory Name: input

Destination File: output1

Encryption Key: •

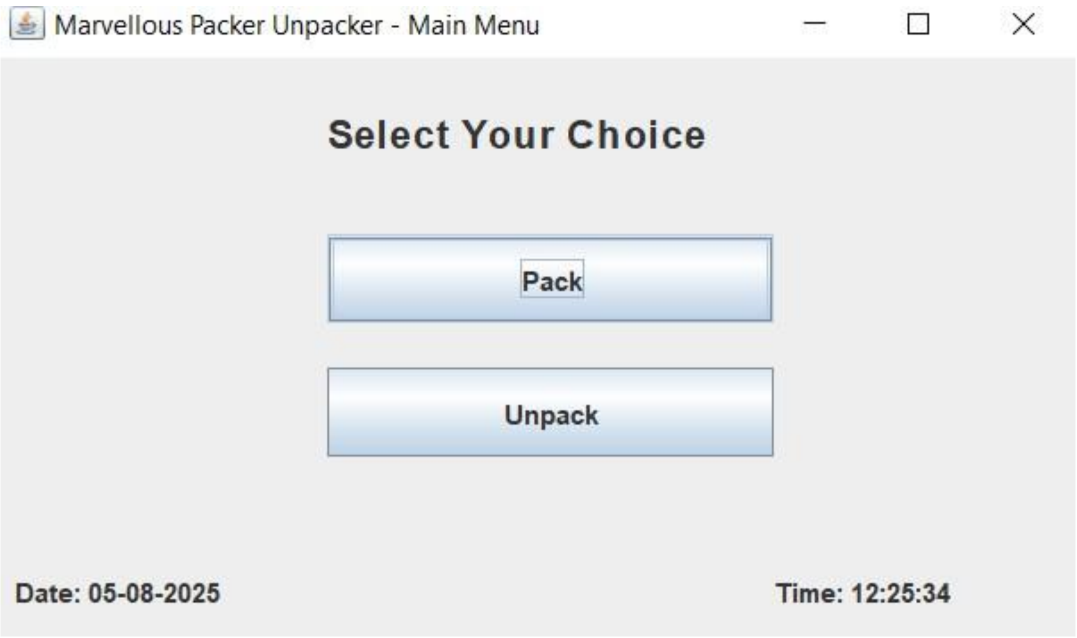
- The user provides the Directory Name, the Destination File name, and an Encryption Key (a required input for securing the files).
- Upon clicking "Pack," the system performs the Packing Activity: it creates the new destination file and populates it with the encrypted metadata (header) and the encrypted data of all files found in the specified directory.
- Once the activity is complete, the user is notified via a message box displaying "Packing Completed!"



Message

Packing Completed!

- After that previous window should be displayed as



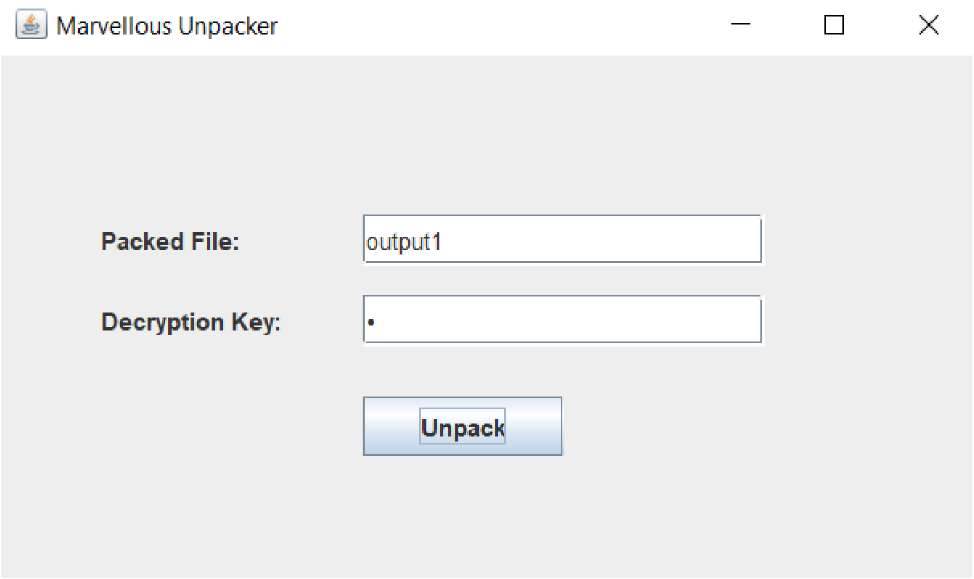
Marvellous Packer Unpacker - Main Menu

Select Your Choice

Date: 05-08-2025

Time: 12:25:34

- When user click Unpack button below window should be displayed as

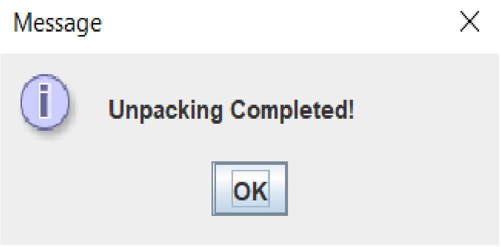


Marvellous Unpacker

Packed File: output1

Decryption Key: •

- From this window accept name of packed file.
- From this window decryption key used for encryption.
- When user click Unpack button we have to perform “Unpacking Activity”.
- When user click Unpack button then Message Box Will Appear As



Unpacking Completed!

Backend Logic

Packing Activity :

The packing process is executed after accepting the source directory name, the destination packed file name, and the Encryption Key from the user via the GUI.

- The utility begins by creating a new regular file with the name specified by the user.
- It then opens the directory and traverses each file within it.
- For every file encountered, the application first generates Metadata (Header), which includes the file's name and its size, ensuring the header is exactly 100 bytes long.
- Encryption is a critical step: the 100-byte header is immediately encrypted using the user-provided Encryption Key before being written to the newly created packed file.
- Following the encrypted header, the actual file data is read in chunks, and encryption is performed on the data using the same Encryption Key before the encrypted data is written sequentially to the packed file.
- Required Feature (Pending): A key project requirement is that each file's name, size, and checksum should be written to a separate log file created in the system directory. This functionality is currently not implemented in the provided code.
- Finally, after completing the directory traversal and processing all files, the application displays a detailed packing report on the console showing the total number of files packed.

UnPacking Activity :

- The unpacking process securely extracts the original files from the consolidated packed file. This process is initiated after the user supplies the packed file name and the corresponding Decryption Key via the GUI

- The utility accepts the packed file name and the essential Decryption Key from the user.
- The system then opens the packed file in read mode and begins the iterative unpacking activity.
- The core extraction loop performs the following sequence of steps until the end of the packed file is reached:

    1. Read & Decrypt Header: A 100-byte header is read from the packed file and immediately decrypted using the provided Decryption Key
    2. File Recreation: The decrypted header is parsed to extract the original file name and its size. A new file is created on the local system using the extracted name
    3. Read & Decrypt Data: The exact number of bytes corresponding to the file size is read from the packed file, decrypted using the Decryption Key, and then written into the newly created file
- Upon completion of all extraction steps, the application displays an unpacking report on the console, providing statistics on the total number of files successfully unpacked