

Project Report
On
Signup Page Using Shell Scripting

MASTERS OF COMPUTER APPLICATIONS



Submitted By:

Harshit Kumar Jaiswal
(24MCA20148)

Project Guide:

Mr. Navdeep Singh Sodhi
MCA Dept., (CU)

DEPARTMENT OF COMPUTER APPLICATION,
CHANDIGARH UNIVERSITY,
(NH05, Chandigarh-Ludhiana Highway, Gharuan, Mohali, Punjab, India)

SESSION 2024-26

DECLARATION

I, Harshit Kumar Jaiswal, hereby declare that this project report titled "*Signup Page Using Shell Scripting*" is original work carried out by me under the supervision of Mr. Navdeep Singh Sodhi. I further declare that this work has not been submitted to any other institute/university for the award of the degree of Master of Computer Applications.

Student Name: **Harshit Kumar Jaiswal**

Roll No: **24MCA20148**

ACKNOWLEDGEMENT

I express my sincere gratitude to my project guide, Mr. Navdeep Singh Sodhi, for invaluable guidance and support throughout this project. I also extend my thanks to Chandigarh University for the opportunity to undertake this project and to my classmates and family for their continuous encouragement.

Harshit Kumar Jaiswal
24MCA20148

TABLE OF CONTENTS

Chapter	Title	Page No.
1	Introduction	5
1.1	Requirements and Tools	5
1.2	System Requirements	5
1.3	Tools and Software	6
2	Design Overview	7
3	Sample Reports	8
3.1	Shell Script Implementation	8
3.2.1	Initial Output	8-10
4	Conclusion and Future Work	11
5	Bibliography	12

Chapter 1: Introduction

This project report aims to explore the development of a signup page using Linux shell scripting. The primary objective is to provide a comprehensive overview of how to create a user-friendly interface that allows individuals to register for services or applications efficiently. By leveraging the capabilities of shell scripting, we aim to demonstrate how this powerful tool can streamline the signup process, making it more accessible and manageable for both developers and users.

A signup page serves as a critical entry point for users to create accounts and access various features of an application or service. It typically includes fields for users to input essential information, such as their name, email address, and password, among other details. The design and functionality of this page are crucial as they significantly impact user experience and engagement. A well-constructed signup page can reduce barriers to entry, encouraging more users to join and participate.

Creating a signup page using Linux shell scripting is particularly relevant in today's tech landscape. Shell scripting provides a lightweight and efficient way to automate tasks that would otherwise be tedious and time-consuming. By employing shell scripts, developers can manage data input, validate user details, and store information securely, all from the command line. This approach not only enhances productivity but also allows for greater customization and control over the signup process. Furthermore, using Linux as the platform for such a project highlights the versatility and power of open-source tools in modern software development, making it an ideal choice for both educational and practical applications.

1.1 Requirements and Tools

To effectively run shell scripts in Linux, certain system requirements and tools are essential. First and foremost, a Linux-based operating system is necessary; popular distributions include Ubuntu, Fedora, and CentOS. Additionally, a minimal installation should include the Bash shell, which is commonly available in most Linux distributions, as it serves as the default command-line interface for executing shell scripts.

1.2 System Requirements:

1. **Operating System:** Any Linux distribution (e.g., Ubuntu, Fedora, Debian).
2. **Processor:** A modern CPU with sufficient processing power (typically any processor from the last decade will suffice).
3. **RAM:** At least 1 GB of RAM is recommended for basic scripting tasks; more may be required for complex operations.
4. **Disk Space:** A minimum of 500 MB free disk space for script files and dependencies.
5. **Network Access:** Internet access may be necessary for downloading packages and dependencies.

1.3 Tools and Software:

In addition to the basic system requirements, several tools can enhance the development and testing of the signup page:

Text Editor: A text editor such as Vim, Nano, or Visual Studio Code is essential for writing and editing shell scripts. Each of these editors has features that can improve the coding experience, such as syntax highlighting and autocomplete.

Version Control System: Using Git for version control allows developers to track changes to their scripts and collaborate more effectively.

Debugging Tools: Tools like ShellCheck can help identify syntax errors in shell scripts and improve script quality by providing suggestions for best practices.

By ensuring that the system meets these requirements and that the necessary tools are in place, developers can create and test an efficient and user-friendly signup page using shell scripting in Linux.

Chapter 2: Design Overview

The architecture of the signup page is a critical component in ensuring a seamless user experience and effective data management. The page is designed to capture essential user information, which includes fields such as username, email address, password, and optional fields like phone number or date of birth. Each input field is accompanied by appropriate labels to guide the user in providing the required information.

User input is collected through a simple form interface, which can be implemented using HTML and styled with CSS for a visually appealing layout. Upon user submission, a shell script processes the input. This script is responsible for validating the data entered by the user. Validation checks include verifying the format of the email address, ensuring the password meets security criteria (e.g., minimum length, inclusion of special characters), and confirming that all required fields are filled out. If any validation checks fail, the user receives immediate feedback, prompting them to correct their entries.

Once the input has been validated, the data is securely stored in a database, such as MySQL or PostgreSQL. The shell script interacts with the database using SQL commands to insert the user's information into a designated table. This process ensures that all user data is organized and retrievable for future reference, particularly for login processes.

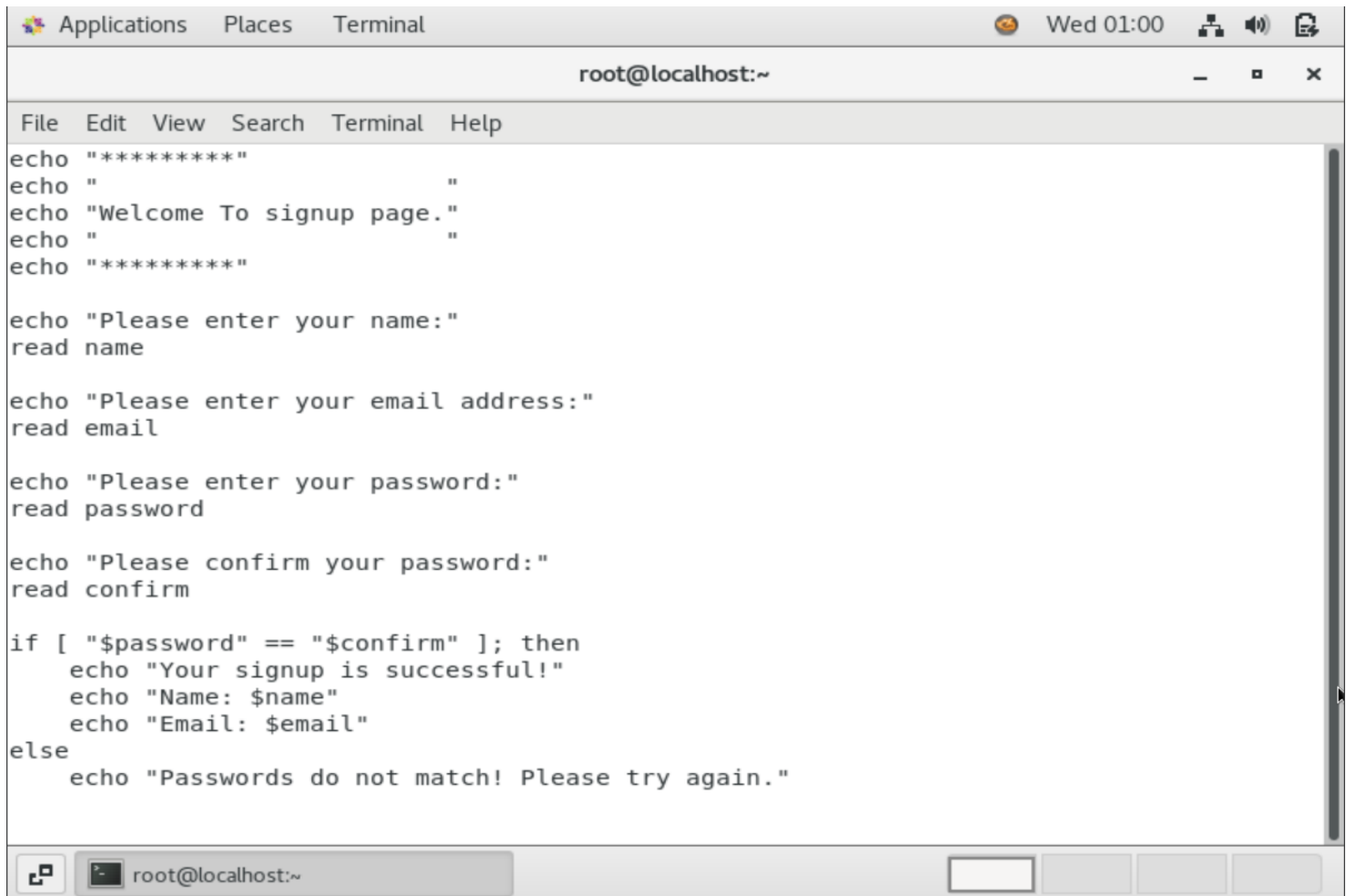
The overall flow from user input to confirmation begins with the user accessing the signup page and filling out the form. Upon submission, the shell script executes the validation checks. If the validation is successful, the user is directed to a confirmation page or message indicating that their registration was successful. This confirmation page can also prompt the user to check their email for verification or provide additional instructions for account activation.

By carefully designing the architecture of the signup page, developers can create a robust system that enhances user experience while ensuring data integrity and security throughout the registration process.

Chapter 3: Sample Reports

3.1 Shell Script Implementation

The implementation of the signup page is executed through a shell script that automates the process of data collection, validation, and storage. Below is a detailed explanation of the shell script code, including comments to clarify the functions and logic used.



```
root@localhost:~  
File Edit View Search Terminal Help  
echo "*****"  
echo " "  
echo "Welcome To signup page."  
echo " "  
echo "*****"  
  
echo "Please enter your name:"  
read name  
  
echo "Please enter your email address:"  
read email  
  
echo "Please enter your password:"  
read password  
  
echo "Please confirm your password:"  
read confirm  
  
if [ "$password" == "$confirm" ]; then  
    echo "Your signup is successful!"  
    echo "Name: $name"  
    echo "Email: $email"  
else  
    echo "Passwords do not match! Please try again."
```

3.2.1 Initial Output:

```
echo "*****"  
echo " "  
echo "Welcome To signup page."  
echo " "  
echo "*****"
```


- These lines are simply printing the welcome message and some decorative lines (***** and spaces) to the screen to make the output more user-friendly.
- The purpose is to greet the user and indicate that they are on a signup page.

3.2.2 Collecting User Input:

```
echo "Please enter your name:"  
read name
```

- The script asks the user to enter their name.
- read name takes the user's input (the name) and stores it in the variable name.

3.2.3 Password Validation:

```
if [ "$password" == "$confirm" ]; then
```

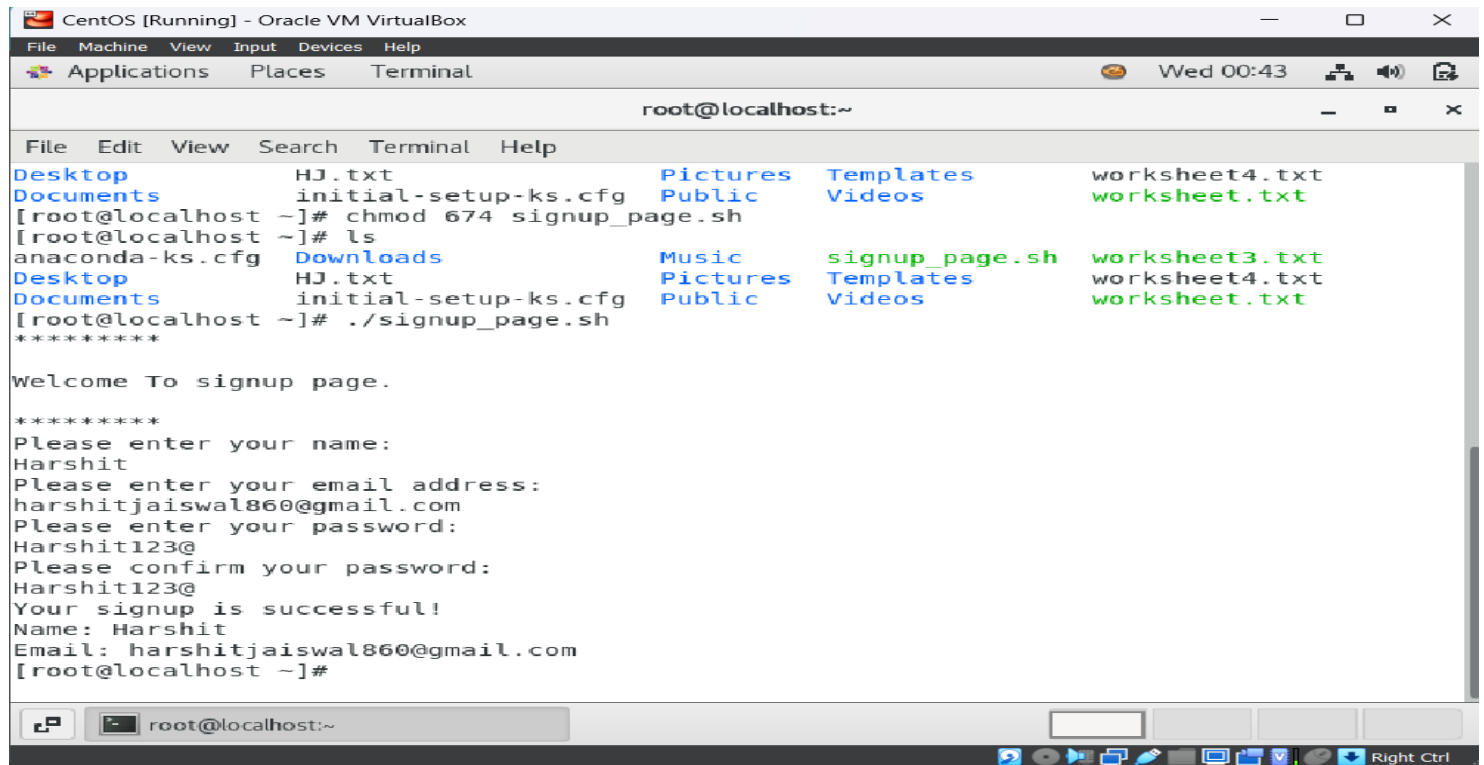
- This line checks if the password entered in the password variable matches the confirmation password entered in the confirm variable.
- The == operator checks if the two values are identical.

3.2.4 Success or Failure:

```
echo "Your signup is successful!"  
echo "Name: $name"  
echo "Email: $email"
```

- If the passwords match, the script prints a message saying the signup is successful.
- It then displays the name and email that the user entered.

Output



CentOS [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Terminal Wed 00:43

root@localhost:~

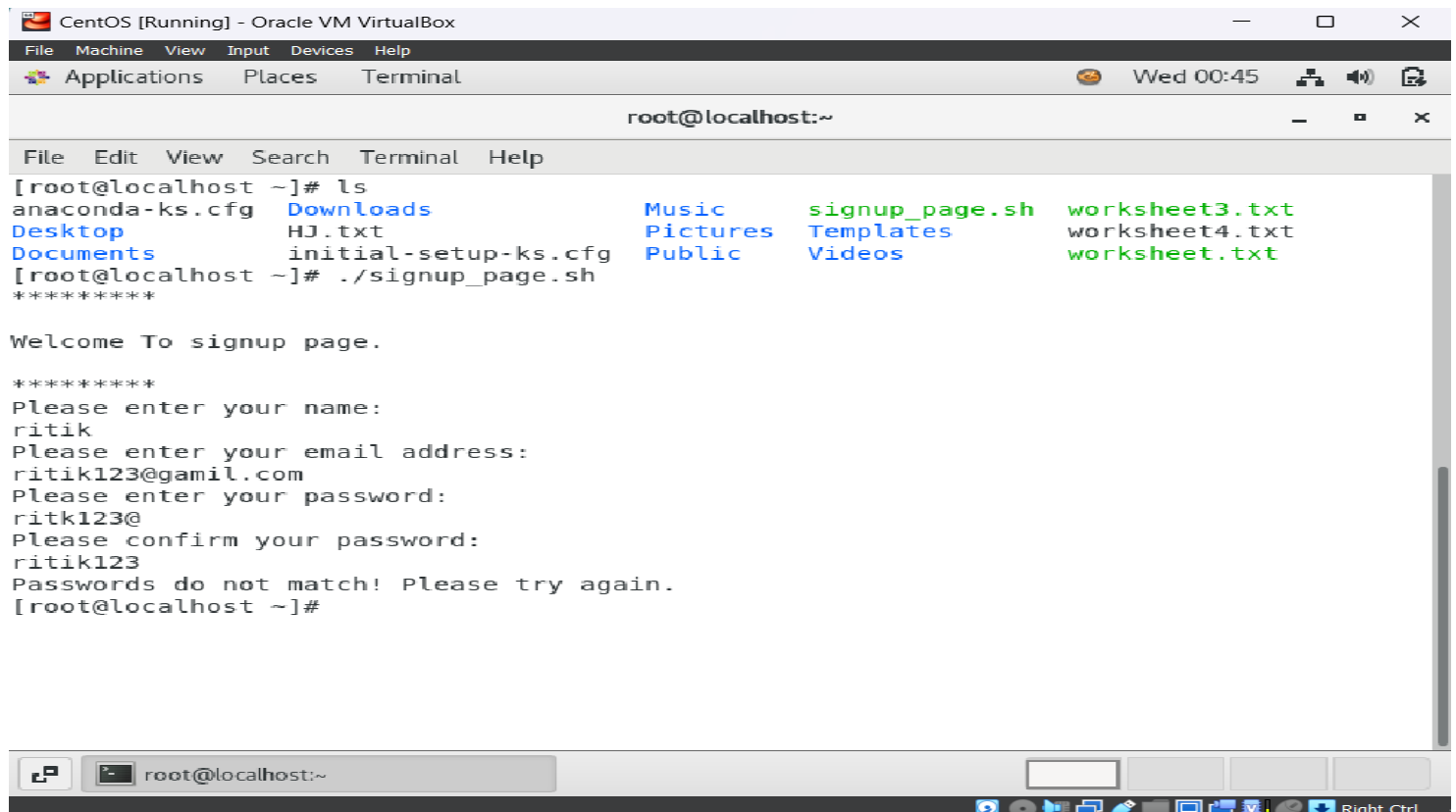
File Edit View Search Terminal Help

Desktop HJ.txt Pictures Templates worksheet4.txt
Documents initial-setup-ks.cfg Public Videos worksheet.txt

```
[root@localhost ~]# chmod 674 signup_page.sh
[root@localhost ~]# ls
anaconda-ks.cfg Downloads Music signup_page.sh worksheet3.txt
Desktop HJ.txt Pictures Templates worksheet4.txt
Documents initial-setup-ks.cfg Public Videos worksheet.txt
[root@localhost ~]# ./signup_page.sh
*****

Welcome To signup page.

*****
Please enter your name:
Harshit
Please enter your email address:
harshitjaiswal860@gmail.com
Please enter your password:
Harshit123@
Please confirm your password:
Harshit123@
Your signup is successful!
Name: Harshit
Email: harshitjaiswal860@gmail.com
[root@localhost ~]#
```



CentOS [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Applications Places Terminal Wed 00:45

root@localhost:~

File Edit View Search Terminal Help

```
[root@localhost ~]# ls
anaconda-ks.cfg Downloads Music signup_page.sh worksheet3.txt
Desktop HJ.txt Pictures Templates worksheet4.txt
Documents initial-setup-ks.cfg Public Videos worksheet.txt
[root@localhost ~]# ./signup_page.sh
*****

Welcome To signup page.

*****
Please enter your name:
ritik
Please enter your email address:
ritik123@gamil.com
Please enter your password:
ritk123@
Please confirm your password:
ritik123
Passwords do not match! Please try again.
[root@localhost ~]#
```

Chapter 4: Conclusion and Future Work:

The development of the signup page using Linux shell scripting has successfully demonstrated a streamlined approach to user registration, emphasizing both functionality and user experience. The project has achieved its primary objectives by providing a secure and efficient method for capturing, validating, and storing user information. Key findings include the effectiveness of using shell scripts for input validation, error handling, and data management in a Linux environment. By leveraging the power of shell scripting, we have created a robust system that enhances accessibility for users while ensuring security protocols are in place for sensitive data. However, there are several areas where future iterations of the signup page could be improved. One potential enhancement is the incorporation of more advanced user feedback mechanisms during the signup process. For instance, integrating real-time validation checks that provide instant feedback as users fill out the form could further improve the user experience. This could involve implementing JavaScript for client-side validation, which operates alongside the existing shell script validations.

Additionally, expanding the database schema to include more user attributes could provide richer user profiles. Features such as user preferences, profile pictures, and social media integration might enhance the signup experience, making it more engaging for users. Furthermore, the implementation of email verification steps post-signup could ensure that users provide valid email addresses, thereby reducing spam registrations and improving data accuracy. Another avenue for future work is the optimization of the performance of the signup page. As the number of users increases, it may be necessary to explore load balancing and caching strategies to handle traffic efficiently. Implementing a more sophisticated web framework in conjunction with the shell script could also provide greater flexibility and scalability for future development.

In summary, while the current implementation of the signup page has proven effective, there are numerous opportunities for enhancement that could lead to an even more effective and user-friendly registration process.

Chapter 5: Bibliography

1. Linux Shell Scripting by Jason Cannon.
2. Linux Programming by Robert Love.
3. Online Resources: Linux.org, StackOverflow and Geeks for Geeks.