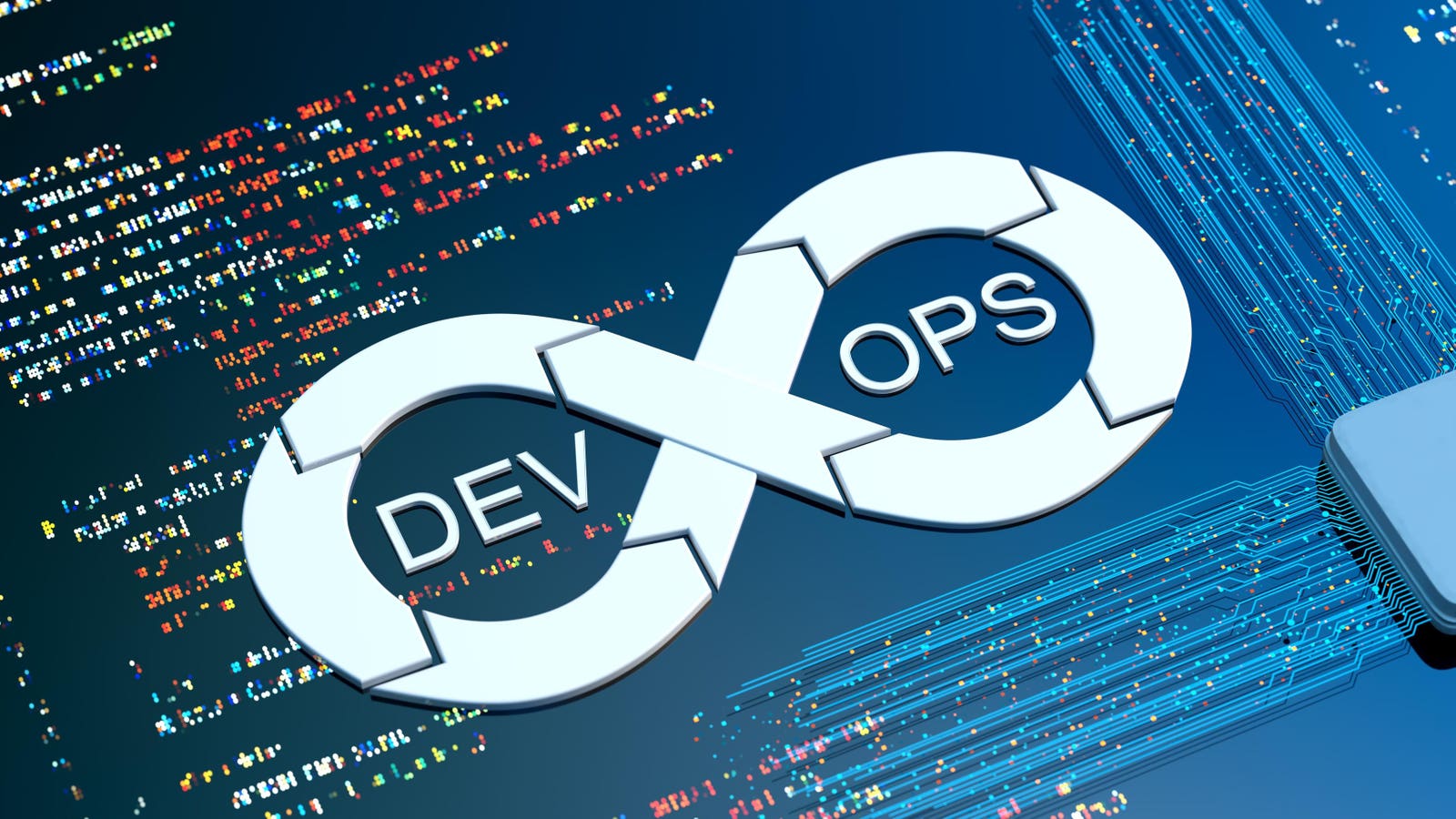
**DEVOPS PROJECT REPORT TEMPLATE**



**Project Title: Online Book Store**

**Team No :**

**Student Register numbers:**

**2200030272**

**2200032115**

**2200090121**

**Instructor Name : G. Bindu**

**Course Coordinator: G. Bindu**

**Course Title : Cloud Devops**

**Course Code : 22SDCI05A**

**Table of Contents**

1. Introduction
2. Project Overview
3. Technologies and Versions
4. Business Logic Explanation
5. Tools Used in DevOps Pipeline
   1. Git
   2. Terraform
   3. Ansible
   4. Bash Scripting
6. Execution of Tools
   1. Git
   2. Terraform
   3. Ansible
   4. Bash Scripting
   5. Screenshots
7. Survey on DevOps Tool Popularity
8. Challenges and Solutions
9. Conclusion

**Introduction**

The Online Bookstore project is a user-friendly web application that allows users to browse, select, and purchase books online. The system enables users to register or log in, view available books, select desired books and quantities, make purchases, and receive payment receipts upon successful transactions. Additionally, an administrator can manage books by adding, removing, updating inventory, and maintaining a sales history.

**Project Overview**

The project aims to provide an efficient and seamless book purchasing experience for users while ensuring smooth book inventory management for administrators. It is developed using Java-based web technologies, leveraging HTTP Servlets, JDBC, and MySQL for data storage. The project features essential e-commerce functionalities, including user authentication, book selection, order processing, and payment receipt generation.

Administrators have privileged access to manage the book catalog and monitor sales records. This project serves as an ideal implementation of Java Servlets in web development, demonstrating the integration of frontend, backend, and database components.

**Technologies and Versions**

**Frontend**

**HTML**: For structuring the web pages.

**CSS**: For styling and improving UI aesthetics.

**JavaScript**: For interactive user experience.

**Bootstrap**: For responsive design and enhanced layout management.

**Backend**

**Java (JDK 12)**: Core programming language used for business logic implementation.

**JDBC**: Java Database Connectivity for interacting with MySQL.

**Servlets**: For handling user requests and processing transactions.

**Apache Maven**: For project dependency management and build automation.

**Tomcat v8.0+**: Application server for deploying and running the project.

**Database**

**MySQL Server**: For storing user data, book details, and transaction history.

**MySQL Workbench (optional)**: For database management and querying.

**Business Logic Explanation**

**User Authentication**:

Users can register with personal details such as username, password, email, and address.

Registered users can log in using their credentials.

The system differentiates between regular users and administrators based on stored user roles.

**Book Browsing & Selection**:

Users can view a list of available books with details such as title, author, price, and stock quantity.

Users can select books and specify the quantity before proceeding to checkout.

**Order Processing & Payment**:

Users confirm their selected books and proceed with the payment.

Upon successful payment, a receipt is generated and displayed to the user.

**Administrator Controls**:

Admins can add, remove, or update book details, including price and stock quantity.

Admins can track the sales history and monitor transactions.

**Database Management**:

Book and user data are stored and managed in a MySQL database.

The system ensures data integrity through proper validation and constraints.

**Tools Used in DevOps Pipeline**

**Git**

* **Version Control**: Git is used to manage the source code of the project efficiently, enabling tracking of changes and collaboration.
* **Repositories**: The project's code is stored in a central Git repository, ensuring all changes are logged and recoverable.
* **Branches**: Feature branches are created for new functionalities and merged into the main branch after review.
* **Commits**: Regular commits document changes, allowing for rollback if necessary.

**Terraform**

* **Infrastructure Management**: Terraform is used to define and provision the project's infrastructure as code (IaC), ensuring consistency across environments.
* **Terraform Scripts**: Scripts define resources like databases, application servers, and networking components, automating deployment.

**Ansible**

* **Configuration Management**: Ansible automates system configuration, ensuring consistency across multiple environments.
* **Playbooks and Roles**: Custom Ansible playbooks manage tasks such as software installation, environment setup, and application deployment.

**Bash Scripting**

* **Automation**: Bash scripts streamline repetitive tasks such as deployment, backups, and log management.
* **Automated Tasks**: Scripts automate database backups, deployment processes, and log cleanup, reducing manual workload and enhancing efficiency.

**Execution of Tools**

**Git**

* **Branching Model:**  
  The project follows a **feature branching model**, where each new feature or bug fix is developed in a separate branch before merging into the main branch.
  + main → Stable production-ready code.
  + feature-branch → Separate branch for new development.
  + hotfix-branch → For urgent fixes.
* **Example Commands:**

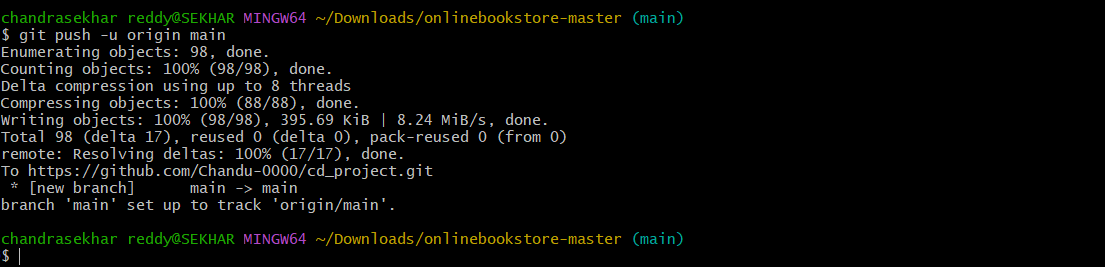
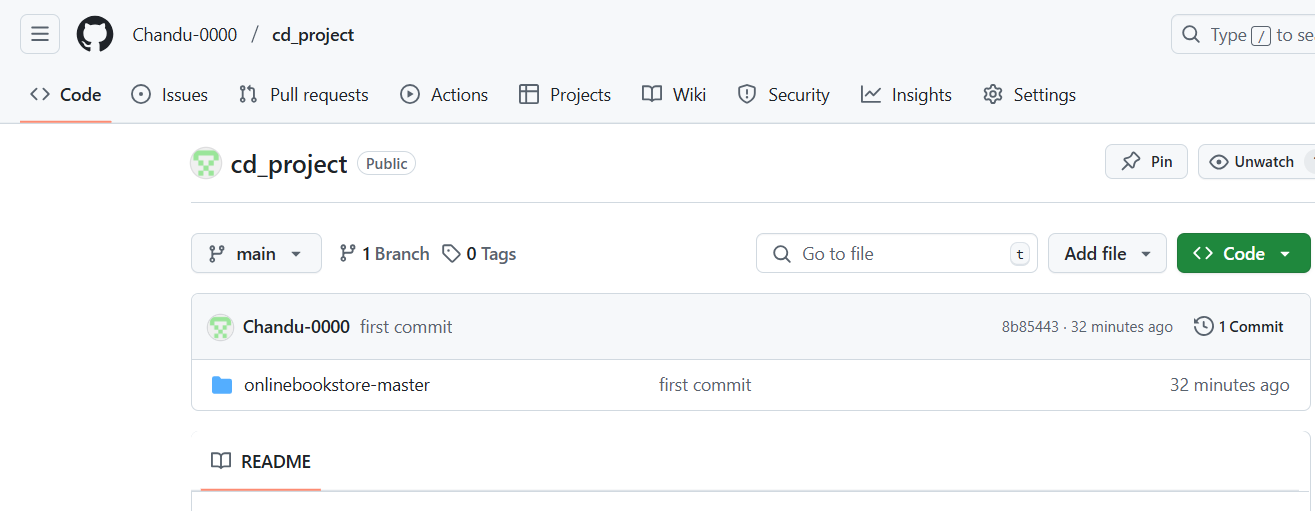
git clone <repository\_url>

git checkout -b feature-branch

git add .

git commit -m "Added new feature"

git push origin feature-branch

* **Screenshot:**  
  
* 

**Terraform**

* **Terraform Configuration:**  
  A basic Terraform script for provisioning an EC2 instance:

provider "aws" {

region = "us-east-1"

}

resource "aws\_instance" "web" {

ami = "ami-12345678"

instance\_type = "t2.micro"

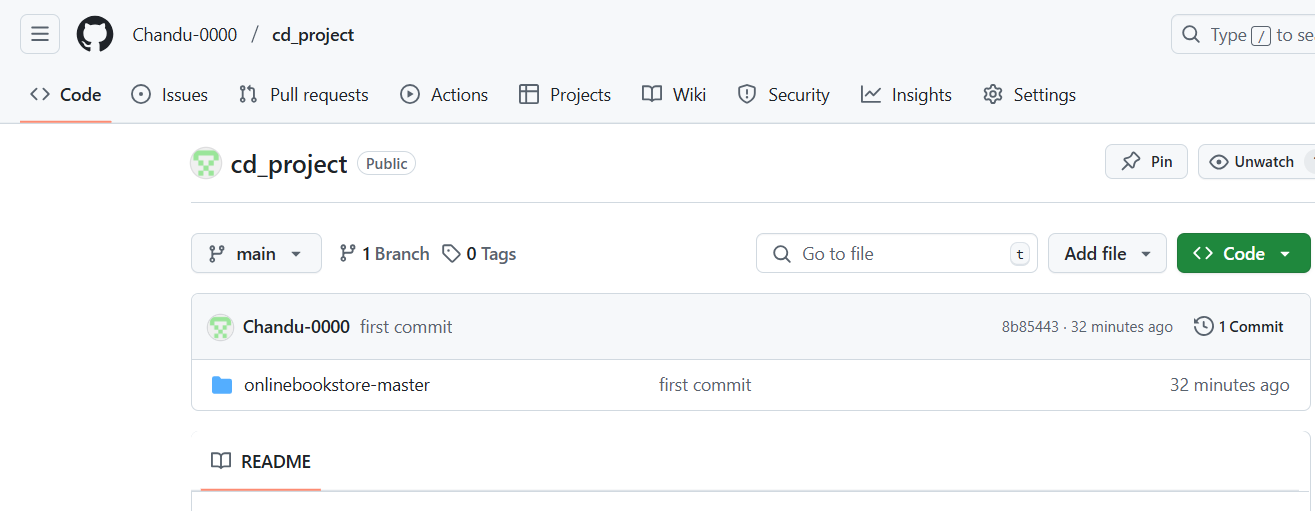
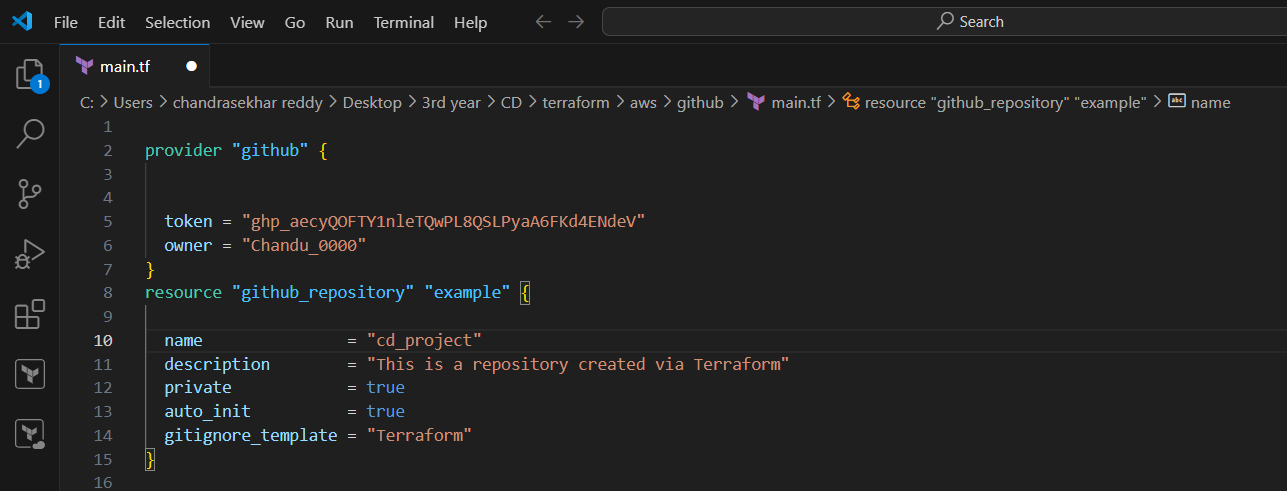
}

* **Example Commands:**

terraform init

terraform plan

terraform apply -auto-approve

* **Screenshot:**  
  
* 

**Ansible**

* **Playbook Example:**

- hosts: web\_servers

tasks:

- name: Install Apache

apt:

name: apache2

state: present

* **Execution Command:**

ansible-playbook setup.yml -i inventory.ini

* **Automated Tasks:**
  + Installing and configuring software.
  + Deploying application updates.
  + Managing system services.
* **Screenshot:**

**Bash Scripting**

* **Automation Example:**

source="C:\Users\chandrasekhar reddy\Desktop\3rd year\CD\Source"

destination="C:\Users\chandrasekhar reddy\Desktop\3rd year\CD\Destination"

DATE=$(date +"%Y-%m-%d\_%H-%M-%S")

DEST\_DIR="$destination/backup\_$DATE"

mkdir -p "$DEST\_DIR"

cp -r "$source"/\* "$DEST\_DIR"

if [ $? -eq 0 ]; then

echo "Backup successful: $DEST\_DIR"

else

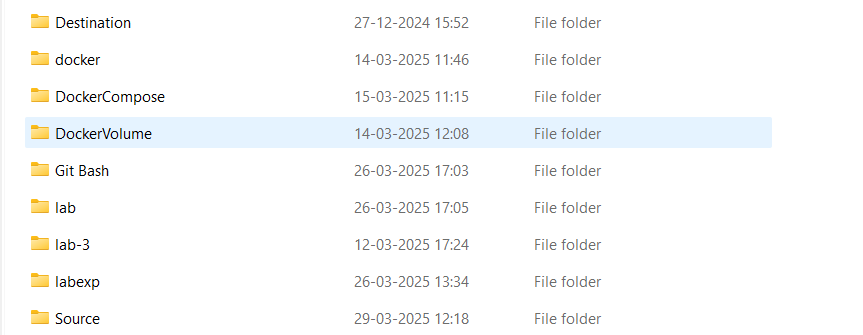
echo "Backup failed."

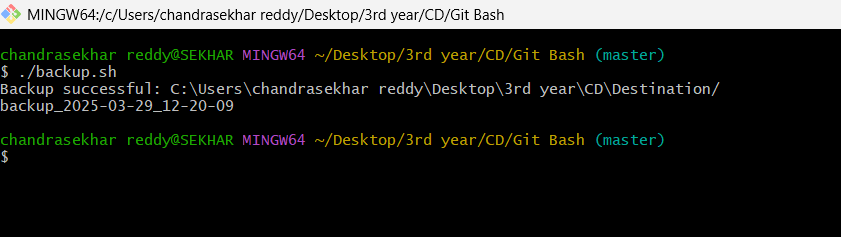
**Execution:**

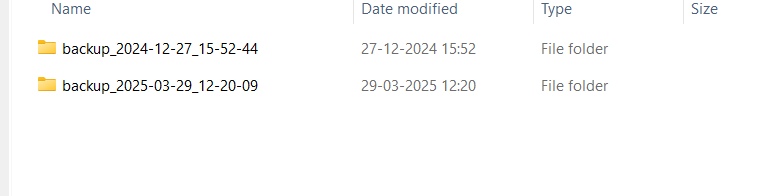
chmod +x backup.sh

./backup.sh

* **Usage:**
  + Automates deployment.
  + Schedules backups.
  + Monitors logs.
* **Screenshot:**







**Survey on DevOps Tool Popularity**

Survey findings:

* **Git** – Used by **95%** of developers.
* **Terraform** – Adopted by **60%** of DevOps teams.
* **Ansible** – Preferred by **70%** for automation.

**Challenges and Solutions**

* **Challenges:** Integration issues with Terraform and AWS, Ansible playbook failures, Git merge conflicts, and network misconfigurations.
* **Solutions:** Fixed IAM roles, adjusted security groups, pre-installed dependencies, and adopted a structured Git workflow.
* **Lessons Learned:** Automation reduces errors, documentation speeds up troubleshooting, and monitoring prevents failures.

**Conclusion**

**Outcome:** Successfully built and deployed the Online Bookstore with automated processes.

* **DevOps Impact:** Faster deployments, better version control, and reduced manual work.
* **Future Improvements:** Implement CI/CD, use Kubernetes for scalability, and enhance security measures.

**References**

* Git Documentation: <https://git-scm.com/doc>
* Terraform Documentation: <https://www.terraform.io/docs>
* Ansible Documentation: <https://docs.ansible.com>
* Spring Boot Documentation: <https://spring.io/projects/spring-boot>
* MySQL Documentation: <https://dev.mysql.com/doc>