



## Assignments 1 and 2: Online Bookstore (10% +10%)

### Objective

In this group assignment, designed for groups of 2 to 3 students, the primary goal is to collectively develop an online bookstore using Python. This platform will empower users to effortlessly browse, search for, and make purchases from a wide selection of books. This project strongly emphasizes teamwork, enhancing programming skills, and mastering the effective utilization of Git and GitHub for streamlined version control.

### Instructions

#### 1. GitHub Repository:

- Create a GitHub repository for the project, and each student should clone the repository to their local development environment.

#### 2. Project Description:

- The group's task is to create an online bookstore application in Python. The bookstore should provide features for browsing books, searching for books by title or author, and purchasing books.

#### 3. Functionality:

- Implement the following features:
  - Catalog: Display a list of books with titles, authors, genres, and prices.
  - Search: Allow users to search for books by title or author.
  - Shopping Cart: Users can add books to a shopping cart and proceed to purchase.
  - User Registration: Implement user registration and login functionality.
  - Purchasing: Users can buy books, and the application should handle payment processing (you can use a simplified payment method for educational purposes).

#### 4. Version Control and Collaboration:

- Divide the tasks among the team members. For instance, one person can work on the catalog, another on the search functionality, and a third on the shopping cart and purchasing.
- Create separate branches for each task.
- Regularly commit and push code to the GitHub repository.
- Use pull requests to merge code into the main branch.
- Conduct code reviews and provide feedback on each other's code through GitHub's review features.

#### 5. Testing:

- Write test cases to ensure that the online bookstore functions correctly. Use Python's testing frameworks, such as `unittest` or `pytest`.

#### 6. Documentation:

- Include a README.md file that provides instructions on how to use the online bookstore, user registration details, and an explanation of the project's structure.

7. Presentation:

- Each group should give a short presentation demonstrating the project's functionality and how GitHub was utilized for version control and collaboration.

8. Grading:

- Evaluate the project based on its functionality, code quality, effective use of Git and GitHub, testing, and presentation.

This assignment encourages you to work as a team and collaborate on a Python project and helps you learn to use version control effectively. It introduces you to practical aspects of web application development and the importance of teamwork in software projects.

<b>Deadline: 24 November 2023</b>				
<b>Final Report and source code to be submitted in GitHub.</b>				
<b>Assignment Progress 1 (10%)</b>				
<b>Assessment Criteria</b>	<b>CLO</b>	<b>Excellent</b>	<b>Moderate</b>	<b>Poor</b>
Knowledge and understanding of the proposed algorithms and data structures are correctly formulated. The proposed solution has been <b>partially</b> implemented and tested.	1 (5%)	4-5 %	2-3 %	0-1 %
Effective use of open source tools in developing the solutions. The collaborative software development environment has been implemented <b>100%</b> .	4 (3%)	3 %	2 %	0-1 %
The project was managed collectively and in an organized manner. Task distributions are excellent. The project is at least <b>30%</b> completed.	5 (2%)	2 %	1 %	0 %
<b>TOTAL</b>	<b>10%</b>			

Deadline: 5 January 2024				
Final Report and source code to be submitted in Github.				
Assignment Progress 2 (10%)				
Assessment Criteria	CLO	Excellent	Moderate	Poor
Knowledge and understanding of the proposed algorithms and data structures are correctly formulated. The proposed solution has been <b>fully</b> implemented and tested.	1 (5%)	4-5 %	2-3 %	0-1 %
Effective use of open source tools in developing the solutions. The collaborative software development environment has been effectively used for debugging and combining different codes from all team members.	4 (2%)	2 %	1 %	0 %
The project was managed collectively and in an organized manner. Task distributions are excellent. The project is <b>100%</b> completed.	5 (3%)	3 %	2 %	0-1 %
<b>TOTAL</b>	<b>10%</b>			

## Deployment (depends on Progress 1)

You can deploy your Python code online using various cloud hosting services, web hosting platforms, or containers. Here's a basic outline of how you can deploy the online bookstore project (or any other web application) created in Python:

### 1. Choose a Hosting Service:

- Select a hosting service that supports Python applications. Some popular options include Heroku, PythonAnywhere, AWS Elastic Beanstalk, and Azure App Service.

## 2. Set Up an Account:

- Create an account on the chosen hosting platform and log in.

## 3. Prepare the Application:

- Ensure the Python application is structured and organized with the necessary files and dependencies. The project should be in a deployable state.

## 4. Deployment Method:

- Different hosting platforms have different deployment methods. Generally, you can deploy your code in one of the following ways:

- Heroku: If using Heroku, you can deploy by connecting your GitHub repository to Heroku and enabling automatic deployments on code pushes.

- PythonAnywhere: With PythonAnywhere, you can upload your project files through the web interface or by using Git.

- AWS Elastic Beanstalk: AWS provides Elastic Beanstalk, a Platform as a Service (PaaS) offering. You can use the AWS CLI or the AWS Management Console to create an Elastic Beanstalk application and upload your code.

- Azure App Service: In Azure, you can create an App Service and deploy your code using Git, FTP, or continuous integration.

## 5. Environment Setup:

- Set up the runtime environment by specifying the Python version and dependencies in a configuration file or through the platform's interface.

## 6. Database and Storage:

- If the application uses a database, configure the database connection. For storage, configure any necessary cloud storage services.

## 7. Test Deployment:

- Deploy the application and test it online. Ensure that it works as expected in the online environment.

## 8. Custom Domain:

- If needed, set up a custom domain for the application. Many hosting services provide guides on how to configure custom domains.

## 9. Security and Scaling:

- Implement necessary security measures, such as securing access to the application, and consider scaling options if the application experiences increased traffic.

## 10. Monitoring and Logs:

- Set up monitoring and logging to keep track of application performance and identify any issues.

11. Backup and Recovery:

- Implement backup and recovery strategies to ensure data integrity.

12. Documentation:

- Document the deployment process, including details of the hosting service used, configurations, and any important information for maintaining the application.

13. Support and Maintenance:

- Be prepared to provide support and maintenance for the deployed application, addressing any issues or updates that may arise.

The specific steps and details may vary depending on the chosen hosting service, but these general guidelines can help you get started with deploying your Python applications online. It is essential to refer to the documentation and resources provided by the hosting service for more detailed instructions specific to the chosen platform.