

Title*:*

*“*SECURE APPLICATION DEVELOPMENT*”*

Program: BSC in Computing (Application Development)

Course: SECURE APPLICATION DEVELOPMENT

Supervisor: Fouzas Konstantinos

Student name: Panagiotis Stamatiou

Student ID: 2119611

Word Count: 2.304

Table of Contents

[Introduction 2](#_Toc164425260)

[Requirements Analysis 3](#_Toc164425261)

[Analysis of the Code 3](#_Toc164425262)

[Basic Libraries: 4](#_Toc164425263)

[Database Design 4](#_Toc164425264)

[Login panel: 4](#_Toc164425265)

[Table of ratings: 4](#_Toc164425266)

[Table of restaurants: 4](#_Toc164425267)

[User Design: 5](#_Toc164425268)

[Implementation: 5](#_Toc164425269)

[Safety: 5](#_Toc164425270)

[Continuous Integration: 5](#_Toc164425271)

[Security Measures 5](#_Toc164425272)

[Encrypt passwords: 5](#_Toc164425273)

[Data Validity Check: 5](#_Toc164425274)

[Protection against CSRF attacks: 5](#_Toc164425275)

[Access Restriction: 5](#_Toc164425276)

[Continuous Completion and Delivery 6](#_Toc164425277)

[Continuous Integration System 6](#_Toc164425278)

[Review process and problem solving 10](#_Toc164425279)

[Conclusions 18](#_Toc164425280)

[The link for the code on GitLab: 19](#_Toc164425281)

[References 19](#_Toc164425282)

# Introduction

Software development is a critical aspect of the modern technological age and offers countless possibilities for creation and innovation. However, the increase in complexity of applications combined with their widespread public acceptance has made it necessary to focus on software security.

This research aims to develop a secure application software, using modern practices and tools that address the growing threats from cybercriminals. The application developed includes a database, a certification mechanism and an access control mechanism, complying with the requirements of the task.

The application also makes allows for scheduling of the tests and it is integrable with Continuous Integration / Continuous Delivery (CI/CD) environments as well as the implementation of static code analysis and security evaluation tools. SonarCloud is a code analysis service. By integrating directly with the CI pipeline or one of the supported DevOps platforms, our code is tested against an extensive set of rules covering many code attributes, such as maintainability, reliability, and security issues in each merge/pull request. In addition, security measures were implemented based on the OWASP security guidelines.

The application developed as part of the research is a system of recommendations for tourism, where users can create an account and log in with their details. The restaurants are then displayed based on the ratings of all users. The app also provides filters for finding restaurants in specific areas and displays restaurants on a map for easier user access.

**Requirements Analysis**

Successful app development starts with a clear understanding of user requirements and the purpose of the app. This section details the functional requirements of our application and how they have been implemented.

* **User Login:** Users have the ability to log in to the app using a user email and password.
* **User Registration:**  Users have the ability to register in the application. All they need is a name, an email and a password.
* **View Restaurants:** Users can see a list of restaurants in the app, along with information about each restaurant based on their ratings.
* **Location Filter:** Users can filter restaurants based on their location by selecting their region.
* **Show Map:** At the end of the list of restaurants you can see all the locations of the restaurants on a map.
* **Security:** The application ensures the security of user data by encrypting the data and protecting against attacks.
* **User Response:** The application provides a quick response to user actions to ensure an excellent user experience.

# Analysis of the Code

## Basic Libraries:

The Python 3 Standard Library is a vast array of modules that you can use for developing various kinds of applications.

This application uses the libraries Flask, Flask-WTF, Flask-Bcrypt, Flask-MySQLdb and Flask-Login to develop a secure and functional web application. The use of these libraries allows the management of registration, logon and secure logout processes of users, as well as interaction with the MySQL database.

Bcrypt is a utility for encrypting files designed for various platforms. The utility uses the encryption blowfish algorithm.

# Database Design

MySQL, launched in 1995, has become the most popular open source database system. The popularity of MySQL and phpMyAdmin has allowed many non-IT specialists to create dynamic websites with MySQL support.

The application database consists of the following tables:

## Login panel:

* + Description: Stores information about users who have registered with the application.
  + Fields:
    - id: User ID number.
    - name: The name of the user.
    - email: The user's email.
    - password: The user's encrypted password.

## Table of ratings:

* + Description: Stores user ratings on restaurants.
  + Fields:
    - id: Evaluation identification number.
    - user\_id: User ID number who made the assessment.
    - restaurant\_id: Restaurant identification number evaluated.
    - rating: The rating given by the user to the restaurant.

## Table of restaurants:

* + Description: Stores information about the restaurants in the app.
  + Fields:
    - id: Restaurant identification number.
    - name: The name of the restaurant.
    - description: A description of the restaurant.
    - location: The location of the restaurant.
    - address: The address of the restaurant.
    - total\_ratings: The total number of reviews.
    - total\_rating\_sum: The sum of ratings.
    - average\_rating: The average rating of the restaurant.
    - latitude: The latitude of the restaurant.
    - longitude: The longitude of the restaurant.
  + The id fields in each table act as primary keys and are unique to each record. There are also some fields that can be used to calculate a restaurant's average rating and total number of reviews.

## User Design:

* + A simple access control mechanism is implemented for the entry and registration of new users.

## Implementation:

* + The app implements the basic paths for entering, registering, logging out, and entering reviews for restaurants.
  + Also, there are validity checks for email and password when registering new users.

## Safety:

* + The application uses secure practices to store passwords, using the bcrypt library for encryption.
  + In addition, there are checks on the validity of data entered by users.

## Continuous Integration:

* + The application is designed and implemented in order to develop a functional and secure web application, using software development best practices.

# Security Measures

Software security is a critical aspect to protect users and their data from various threats. Below are the security measures that have been implemented in the application:

## Encrypt passwords:

* + User passwords are encrypted before being stored in the database using the Bcrypt library. This ensures that even if the database is leaked, the codes remain secure.

## Data Validity Check:

* + Before saving to the database, the validity of the data entered by users is checked. For example, the format of the email and password is checked.

## Protection against CSRF attacks:

* + A secure CSRF (Cross-Site Request Forgery) mechanism is used using the Flask-WTF library.

## Access Restriction:

* + Some activities, such as viewing profiles or registering reviews, are protected from unauthorized access through Flask-Login's access control mechanism.

The above security measures are implemented in the application in order to provide maximum level of protection for its users.

## Continuous Completion and Delivery

This module demonstrates the use of continuous integration (CI) and delivery (CD) systems to automate the code development and testing process using SonarCloud and GitLab.

## Continuous Integration System

* CI/CD tool: The CI/CD process was implemented using GitLab CI to automate the completion and delivery process.
* SonarCloud integration: SonarCloud was used to continuously evaluate code quality and produce quality reports.
* CI/CD settings: Defined environment variables such as SONAR\_USER\_HOME and GIT\_DEPTH, as well as sonarcloud-check to check the code in SonarCloud.

Image containing text, screenshot, software, computer icon

Auto-generated descriptionImage containing text, screenshot, software, computer icon

Auto-generated descriptionImage containing text, software, screenshot, computer

Auto-generated descriptionImage containing text, software, screenshot, computer

Auto-generated description

CI/CD process: Code changes are automatically reviewed with each commit using GitLab CI, while quality reports are automatically generated by SonarCloud.

Image containing text, screenshot, software, computer icon

Auto-generated descriptionImage containing text, screenshot, software, computer icon

Auto-generated description

# Review process and problem solving

It appears that the audit found a Bandit security issue found in the code. This issue is related to running a Flask application with debug=True, which exposes the Werkzeug debugger and can potentially allow arbitrary code to run.

Image containing text, software, computer icon, web page

Auto-generated description To fix this issue, we should avoid running the Flask debug=True application in a production environment. Debugging can expose sensitive information and should only be used in development. We can set the debug parameter to False when running the Flask application in production:

Image containing text, screenshot, software, media software

Auto-generated description

Using remote artifacts without integrity checks can lead to malicious code execution in the application unexpectedly.

On the client side, where front-end code is running, malicious code could:

1. impersonating users' identities and exploiting their privileges in the application.
2. Add silent malware that monitors users' session and logs sensitive secrets.
3. Access sensitive customers' personal data.
4. alter or otherwise affect the general availability of the app.
5. Mine cryptocurrencies in the background.

Similarly, a compromised piece of software deployed in a server-side application could badly affect the security of the application. For example, server-side malware could:

1. Access and modification of sensitive technical and business data.
2. increase its privileges on the underlying operating system.
3. Use the compromised application as an axis to attack the local network.

By ensuring that a remote artifact is exactly what it is supposed to be before using it, the app is protected from unexpected changes applied to it before downloading it. Especially, integrity checks will allow the detection of an artifact that was replaced by malware on the publishing site, or that was legally changed by its author, in a more benign scenario.

Image containing text, screenshot, software, computer icon

Auto-generated description

Recommended secure coding practices

To check the integrity of a remote artifact, hash verification is the most reliable solution. It ensures that the file has not been modified since the fingerprint was calculated.

The src parameter specifies the URL from which the JavaScript script will be loaded.

The integrity parameter contains the SHA-384 hash of the JavaScript script content. This is used to ensure that the script we load is what is expected and has not been modified between requests. If the hash of the downloaded content does not match the default hash, the browser will not run the script and will display an error.

The async and defer attributes are used to delay script execution until the entire HTML page is loaded, which can improve page performance.

Image containing text, screenshot, software, media software

Auto-generated description Because it is easy to extract strings from an application source code or binary, the credentials should not be encoded. This is especially true for distributed or open source applications.

Credentials must be stored outside of code in a configuration file, database, or secrets management service.

This rule highlights instances of hard-coded credentials that are used in database and LDAP connections. Looks for hard-coded credentials in connection strings and variable names that match any of the patterns from the provided list.

Image containing text, screenshot, software, computer icon

Auto-generated description

I made an .env file containing the MY\_SECRET\_KEY then I can use the python-dotenv package to load these environment variables from the .env file into the Flask application.

Next, I use dotenv to load the environment variables from the .env file into the Flask application.

This way, the MY\_SECRET\_KEY will be loaded from the .env file and used as the secret key for the Flask application.

Image containing text, screenshot, software, media software

Auto-generated description

A cross-site request spoofing (CSRF) attack occurs when a trusted user of a web application can be forced, by an attacker, to perform sensitive actions that they did not intend, such as updating their profile or sending a message, anything that can change the state of the application.

The attacker can trick the user/victim into clicking on a link, corresponding to the privileged action, or visiting a malicious website that embeds a hidden web request, and as web browsers automatically include cookies, actions can be authenticated and sensitive.

Image containing text, software, computer icon, web page

Auto-generated description

The Cross Site Request Forgery (CSRF) method is a penetration technique aimed at exploiting website security vulnerabilities and there is one tool commonly used to find security vulnerabilities on websites, namely OWASP ZAP.

By initializing CSRFProtect with the Flask (csrf.init\_app(app)), you ensure that our application is protected from CSRF attacks. CSRFProtect is an extension for Flask that provides protection against Cross-Site Request Forgery (CSRF) attacks for our app. In this way, you ensure that our application is protected from these types of attacks, which may allow unauthorized actions to be performed on behalf of an authorized user.Image containing text, screenshot, software, media software

Auto-generated descriptionImage containing text, screenshot, software, media software

Auto-generated descriptionImage containing text, screenshot, software, media software

Auto-generated description

Define a constant instead of duplicating this literal 'Register.html' 4 times.Image containing text, screenshot, software, computer icon

Auto-generated description

We define a constant to store the name of the template file and then use that constant throughout your code.

Image containing text, screenshot, software, media software

Auto-generated description

Next, we replace all instances of the string 'Register.html' with the constant REGISTER\_TEMPLATE.Image containing text, screenshot, software, media software

Auto-generated description

Image containing text, software, computer icon, web page

Auto-generated description

# Conclusions

Concluding from the review of the security of the application, we observe the existence of important measures to protect data and users. Password encryption using Bcrypt offers reliable protection against unwanted access to user accounts. In addition, checking the correctness of the email and password during registration enhances the security of users' input data.

The CSRF (Cross-Site Request Forgery) measure used additionally protects app forms from cross-site request spoofing attacks. This ensures that only requests coming from the same website can be executed, thus protecting the application from attacks.

Image containing text, screenshot, software, computer icon

Auto-generated description

# The link for the code on GitLab:

https://gitlab.com/PsAvatar1/SECURE-APPLICATION-DEVELOPMENT.git

# References

Anon., 2023. *What is CI/CD?.* [Online]   
Available at: https://www.redhat.com/en/topics/devops/what-is-ci-cd

Ashutosh Kumar Sinha, S. T., 2019. *CookieArmor: Safeguarding against cross-site request forgery and session hijacking.* [Online]   
Available at: https://onlinelibrary-wiley-com.ezproxy.bolton.ac.uk/doi/full/10.1002/spy2.60

Bell, C., 2022. *MySQL Database Service Revealed: Running MySQL as a Service in the Oracle Cloud Infrastructure.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/mysql-database-service/9781484289457/?ar=&email=u0KPjMumntWAh2Q%2F1HtmBRg9cl6poWxc&tstamp=1713519296&id=254F853F0D66AB52D22F389CB3BA0C345B9AEC3E

Christopher Cowel, N. L. C. T., 2023. *Automating DevOps with GitLab CI/CD Pipelines.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/automating-devops-with/9781803233000/B18073\_FM.xhtml

Delisle, M., 2006. *Creating your MySQL Database: Practical Design Tips and Techniques.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/creating-your-mysql/9781904811305/?ar=&orpq=&email=u0KPjMumntWAh2Q%2F1HtmBRg9cl6poWxc&tstamp=1713519433&id=B741F37CBD4CD70BA5058DE3A7090801C1DDB8A4

GitLab, n.d. *Get started with GitLab CI/CD.* [Online]   
Available at: https://docs.gitlab.com/ee/ci/

Hellmann, D., 2017. *The Python 3 Standard Library by Example, Second Edition.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/the-python-3/9780134291154/

Ilham Firman Ashari, V. O. R. G. S. S. D., 2022. *Analysis of Cross Site Request Forgery (CSRF) Attacks on West Lampung Regency Websites Using OWASP ZAP Tools.* [Online]   
Available at: https://doaj.org/article/c6ccaf3674384f15b939208b1401fcb7

Jun Wang, W. L. K.-F. C., 2016. *Learning to Hash for Indexing Big Data—A Survey.* [Online]   
Available at: https://ieeexplore-ieee-org.ezproxy.bolton.ac.uk/document/7360966

Krogh, J. W., 2018. *MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/mysql-connector-python-revealed/9781484236949/?ar=&orpq=&email=u0KPjMumntWAh2Q%2F1HtmBRg9cl6poWxc&tstamp=1713514084&id=F8196F1747B40A04FC198947E9C2AB2AC79A52C0

Lott, S. F., 2018. *Functional Python Programming - Second Edition.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/functional-python-programming/9781788627061/

Molina, A., 2018. *Modern Python Standard Library Cookbook.* [Online]   
Available at: https://learning-oreilly-com.ezproxy.bolton.ac.uk/library/view/modern-python-standard/9781788830829/?ar=&orpq=&email=u0KPjMumntWAh2Q%2F1HtmBRg9cl6poWxc&tstamp=1713513418&id=865232EEB92882D4F0733BBFE830C7C1F2AA57B9

Paul, S. & Singh, A., 2020. *Hands-On Python Deep Learning for the Web.* [Online]   
Available at: https://prism.librarymanagementcloud.co.uk/bolton-ac/items/342198

Shelley, J., n.d. *Bcrypt.* [Online]   
Available at: https://www.solvusoft.com/el/file-extensions/software/johnny-shelley/bcrypt/

SonarCloud, n.d. *SonarCloud Documentation.* [Online]   
Available at: https://docs.sonarsource.com/sonarcloud/

Zhygulskyy, M., 2021. *Automated, Scheduled and Ci /Cd Web Injection.* [Online]   
Available at: https://www.proquest.com/docview/2586971494/fulltextPDF/159847D72E2C4A57PQ/1?accountid=9653&sourcetype=Dissertations%20&%20Theses