

# Cover Page

---

## Title of the Laboratory Work:

PYTHON FRAMEWORKS - Test 1: Variant 1

## Description:

Input a positive integer number and check if this number belongs to the Fibonacci series.

## Student's Name:

Ahmet Enes Dur

## Student Group:

KH-221ia.e

## Instructor's Name:

Svitlana Kovalenko

## Institution's Name:

National Technical University "Kharkiv Polytechnic Institute"

## Department/Faculty:

Department of Software Engineering and Management Intelligent Technologies

- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as
- as

- as

# Report on the Fibonacci Checker Web Application

---

## Introduction:

The objective of this project was to craft a straightforward web application utilizing the Flask framework to ascertain if a given positive integer is a member of the Fibonacci sequence.

---

## Requirements:

1. Craft the application utilizing the Flask framework.
  2. The application must have distinct pages for input and output.
  3. Data entry should be facilitated by Flask-WTF.
  4. Validate the input: Disallow empty strings, non-numbers, and invalid number types like floating points.
  5. Present the outcome through flash messages.
  6. Besides the result, exhibit a related image (either a 'yes' or 'no' image).
  7. Integrate Bootstrap classes along with tailor-made CSS styles.
- 

## Implementation Steps:

1. **Setting up the Application Structure:** The application's files and directories were neatly organized for optimal readability and scalability.

```
/test_1
├── /static
│   ├── /css
│   │   └── styles.css
│   └── /images
│       ├── yes.png
│       └── no.png
├── /templates
│   ├── index.html
│   └── result.html
├── /venv
└── app.py
```

2. **Designing the Input Form:** Flask-WTF was used to craft a simple form, comprising an integer field and a submit button.
3. **Input Validation:** The application strictly accepts positive integers. Any other form of input is flagged, ensuring the validity of the results.

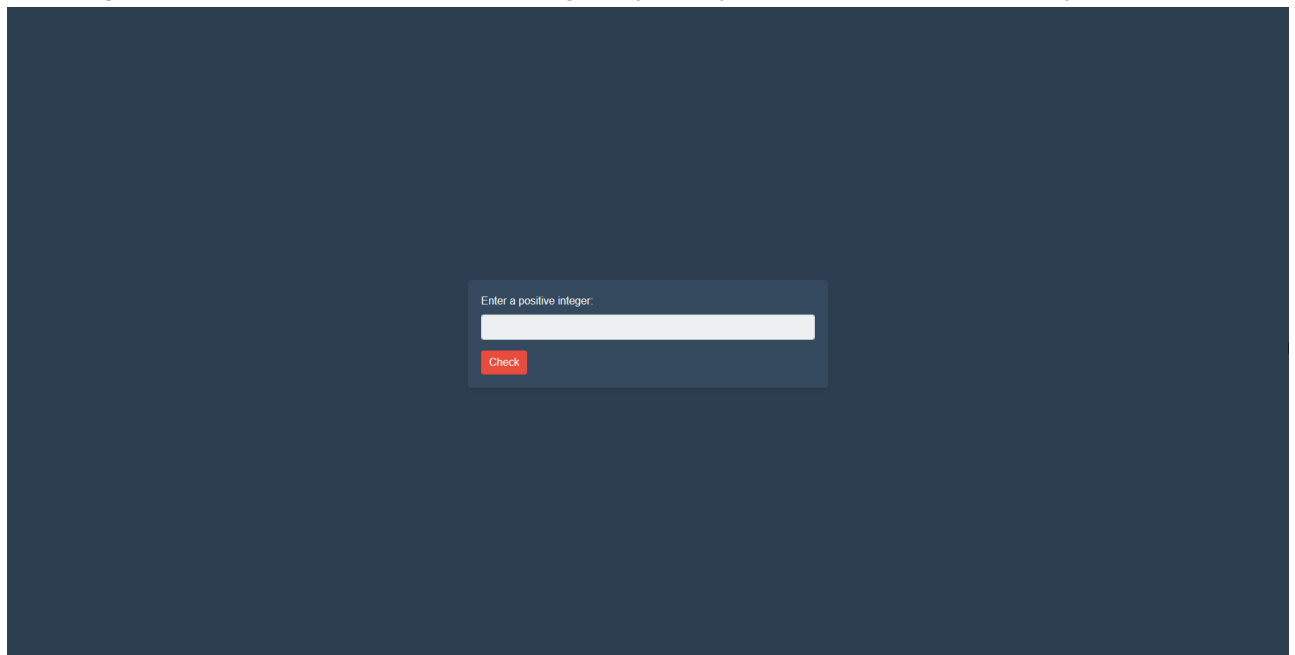
4. **Fibonacci Sequence Logic:** A dedicated Python function was used to verify if a number resides within the Fibonacci sequence.
  5. **Presenting the Results:** Once processed, the outcomes are showcased via flash messages, a native feature of Flask. Depending on the result, a relevant image is displayed adjacent to the message.
- 

## Code and Libraries:

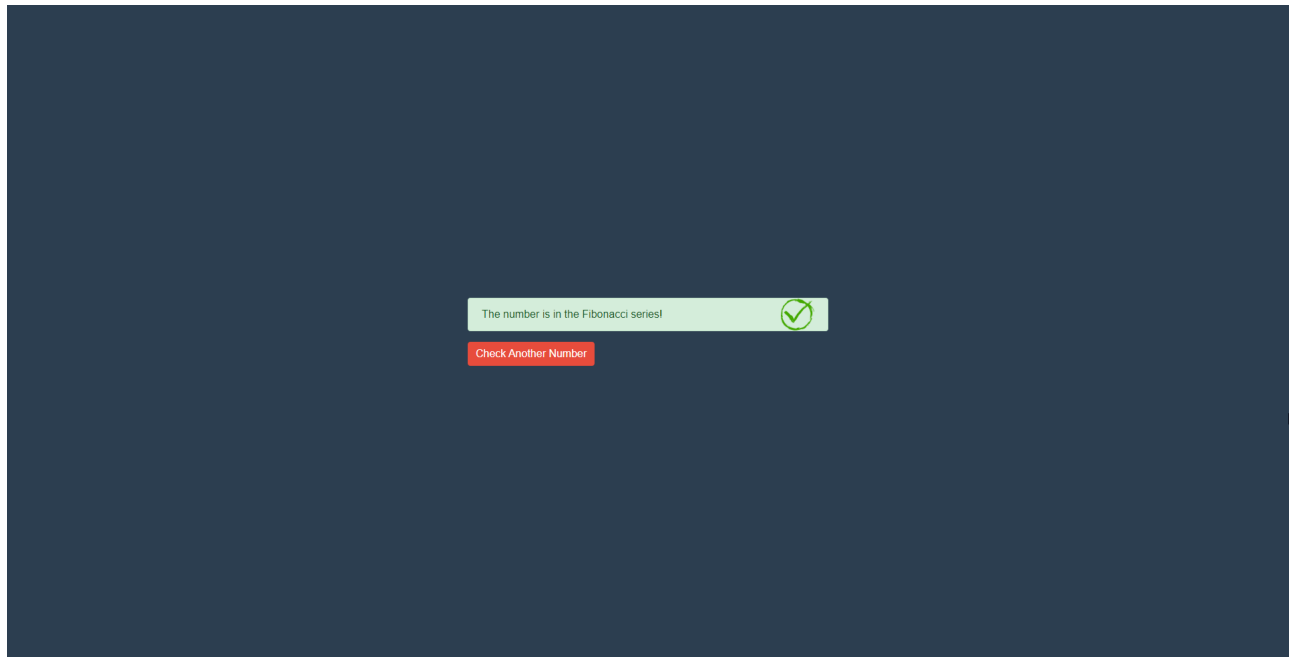
1. **Flask:** A streamlined web application framework, which was the backbone of our web application.
  2. **Flask-WTF:** A Flask extension that smoothly integrates the WTForms module, pivotal for crafting our data input form and managing form submissions.
  3. **Bootstrap:** Employed for swift and adaptable design. Numerous Bootstrap classes were used for both the form and the display.
  4. **Custom CSS:** Found within `static/css/styles.css`, this document holds the application's dark-themed aesthetics. The design relies on Flexbox for content centering and other CSS techniques for visual enhancement.
- 

## Screenshots:

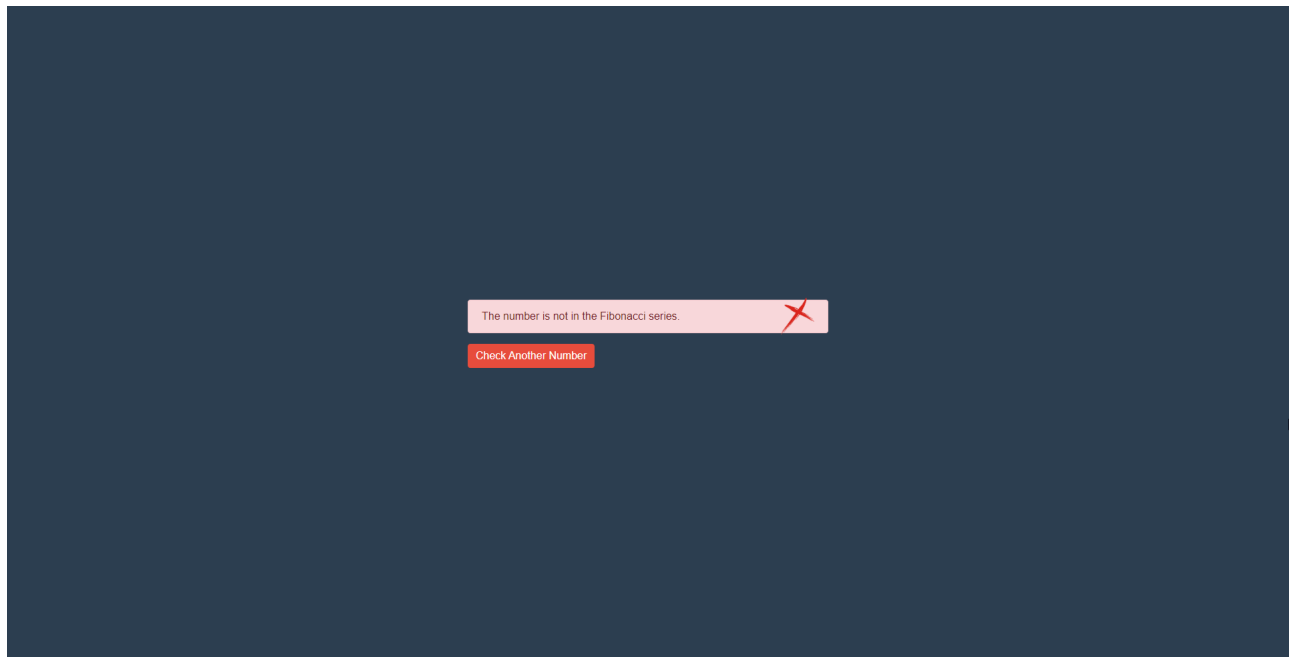
1. **Main Input Screen:** A screenshot showcasing the primary screen where users can input a number.



2. **Positive Outcome:** An image that demonstrates a positive outcome with the corresponding "yes" image.



3. **Negative Outcome:** A display snapshot that highlights a negative result equipped with the "no" image.



---

## Conclusion:

The Fibonacci Checker web application was flawlessly realized, adhering to the prescribed requirements. The amalgamation of the Flask framework, Flask-WTF, and Bootstrap ensured a seamless development journey. The final product is intuitive, adaptable, and precisely verifies Fibonacci numbers.