C3F1

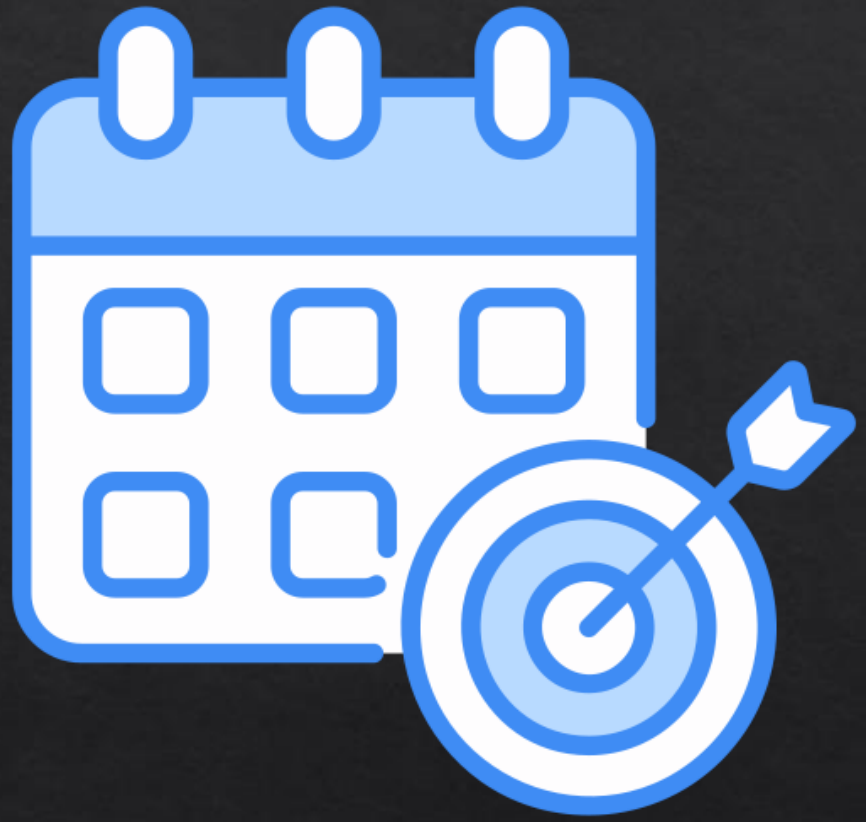
Project proposed by:

Calò Domenico, Capuano Marzia, Crispino Luca, Ficarella Fabrizio

A.A. 2024/2025

MAIN OBJECTIVE

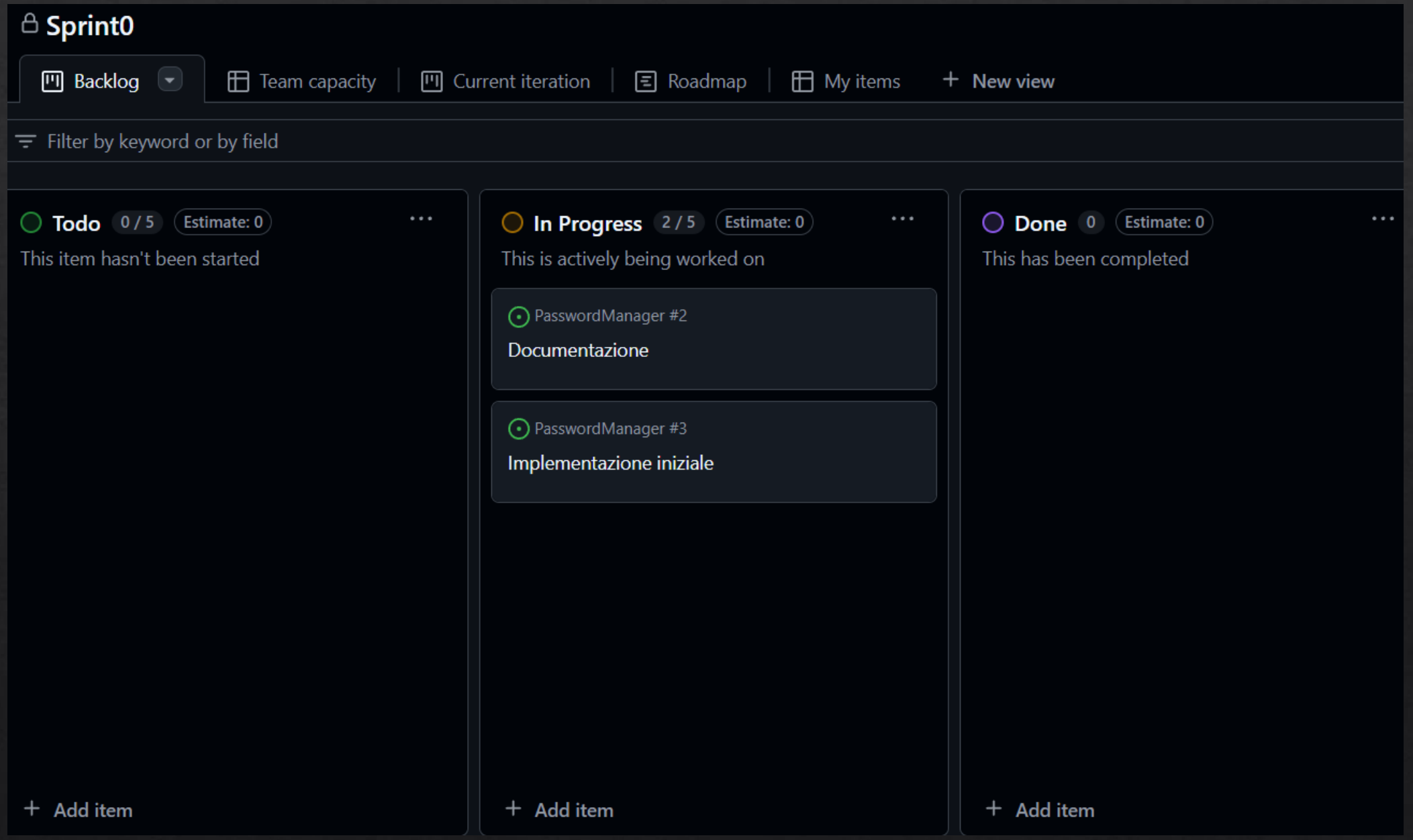
The project consists of developing a secure and user-friendly Password Manager designed to help individuals and organizations store, manage, and generate strong passwords and other secrets.



CHALLENGES

- Effective encryption methods
- Password Generation
- Accessible everywhere
- Stores all types of sensitive data

Github


Sprint0
Backlog
Team capacity
Current iteration
Roadmap
My items
New view
Filter by keyword or by field
Todo 0 / 5 Estimate: 0
This item hasn't been started
Add item
In Progress 2 / 5 Estimate: 0
This is actively being worked on
PasswordManager #2
Documentazione
PasswordManager #3
Implementazione iniziale
Add item
Done 0 Estimate: 0
This has been completed
Add item

SPRINT'S PROGRAM

0 → 1 → 2

DOCUMENTATION: Documentation and analysis of the current state of the art, to see the shortcomings and possible implementations of a password manager project and decide the features to be implemented.

IMPLEMENTATION: Evaluation of the technologies that can be used and verification of a possible synergy between them, verifying that the use of them is as expected.

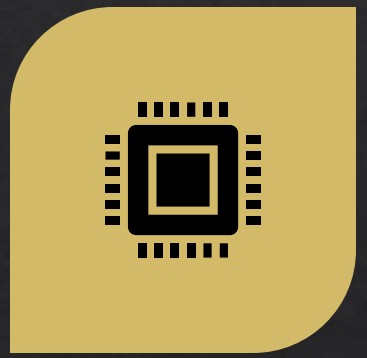
CODE IMPLEMENTATION: Creation of the password manager program, integrating the features we want, through the drafting of the code on the backend side, on the frontend side.

DATABASE CREATION: Creation of the database, integrating it with the encryption mechanism.

DEBBUGING: We carry out the process of identifying, analyzing and fixing errors within our software. During this phase, we carry out problem solving of unexpected malfunctions.

REPORT: Drafting of the end-of-project report using LaTex.

PROPOSED SOLUTION



FOR AN OPTIMAL SOLUTION, IT WAS DECIDED TO USE A CRYPTOGRAPHY BASED ON THE PYTHON'S **FERNET** LIBRARY THAT ALLOWS DIFFERENT TYPES OF EFFECTIVE SOLUTIONS SUCH AS THE STANDARD AES 128.

THE IDEA IS TO GENERATE AT FIRST USE OF THE APP A **MASTER KEY** THAT ALLOWS THE USER TO ENCRYPT AND DECIPHER A FILE OR ENCRYPTED DATA CONTAINING SENSITIVE INFORMATION.

```python
from flask import Flask, request, render_template, redirect, url_for
from cryptography.fernet import Fernet
import os
import json

app = Flask(__name__)

# Percorsi ai file
KEY_FILE = 'master_key.key'
PASSWORD_FILE = 'passwords.enc'

# Funzione per generare e salvare la master key
def generate_master_key():
    key = Fernet.generate_key()
    with open(KEY_FILE, 'wb') as key_file:
        key_file.write(key)
    return key

# Funzione per caricare la master key
def load_master_key():
    if os.path.exists(KEY_FILE):
        with open(KEY_FILE, 'rb') as key_file:
            return key_file.read()
    return generate_master_key()
```

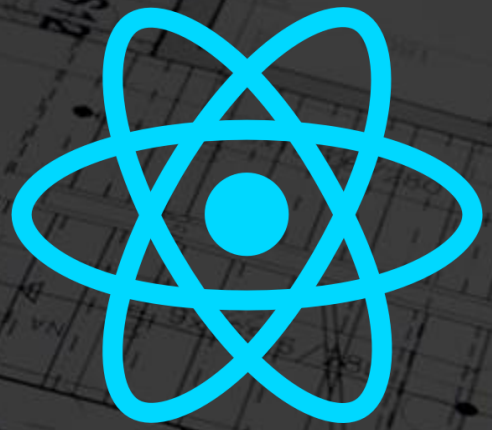
SOFTWARE ARCHITECTURE

To create our password manager we'll use principally:

- Django or Flask for the backend
- React for the frontend

Possible changes can be done to the structure.

Flask



REQUIREMENTS

FUNCTIONAL

- Password generator
- TOTP
- Auto-fill
- Secure storage
- Cross-platform

NON-FUNCTIONAL

- Security
- Reliability
- Portability
- Usability

A.A. 2024/2025