

# **PORTFOLIO**

Create a Habit Tracking App

Finalization Phase/Phase 3

## **Abstract**

Name: Oluwatoyin Eniola Ipadeola

Mat Number: 92130758

Course Code: DLBDSOOFPP01

Course: Object-Oriented And Functional Programming With Python

Submission Date: 10/22/2025

Repository: [https://github.com/ipadeolaoluwatoyin7880/habit\\_tracker](https://github.com/ipadeolaoluwatoyin7880/habit_tracker)

TUTOR: Mirmehdi Seyedebrahim

# PROJECT ABSTRACT: HABIT TRACKER APPLICATION

## PROJECT OVERVIEW

This habit tracking application represents a comprehensive implementation of modern Python programming paradigms, combining Object-Oriented Programming (OOP) principles with Functional Programming (FP) techniques. The project successfully delivers a production-ready application that enables users to define, manage, and analyze their habits.

## TECHNICAL APPROACH & ARCHITECTURE

### Layered Architecture Design

The application follows a clean architecture pattern with clear separation of concerns across five distinct layers:

- Presentation Layer (CLI): Interactive command-line interface using Questionary
- Business Logic Layer (Managers): Workflow coordination and business rules
- Domain Layer (Data Model): Core business entities and relationships
- FP Analytics Layer (Analytics): Pure functional programming for analytics
- Data Persistence Layer (Storage): Database abstraction and management

### Key Technical Decisions

- SQLite Database: Chosen for its simplicity, reliability, and zero-configuration requirements
- Questionary Library: Selected for creating an intuitive, user-friendly CLI experience
- Pytest Framework: Implemented for comprehensive test coverage and CI/CD readiness
- SHA-256 Hashing: Used for secure password storage with salt addition

## SUCSESSES & ACHIEVEMENTS

### What Went Well

- Clean Code Architecture: The layered approach proved highly maintainable and testable
- Comprehensive Testing: 40+ tests covering all critical paths ensured reliability
- FP Implementation: Pure functions for analytics provided predictable, bug-free calculations
- OOP Design Patterns: Factory, Repository, and Service Layer patterns created flexible, extensible code

### Technical Innovations

- Polymorphic Habit System: Daily and weekly habits with different periodicity logic
- Streak Calculation Algorithms: Efficient algorithms handling edge cases like year boundaries
- Soft Deletion: Data preservation while maintaining application integrity

- Windows Compatibility: Robust file handling and connection management

## **CHALLENGES & SOLUTIONS**

### **Unexpected Challenges**

- DateTime Handling: Complex timezone and boundary calculations for streak tracking  
Solution: Implemented ISO week-based tracking with proper boundary handling
- Database Connection Management: Ensuring proper cleanup on Windows systems  
Solution: Context managers and graceful degradation strategies
- FP Purity Maintenance: Keeping analytics functions truly side-effect free  
Solution: Strict immutability and referential transparency enforcement

### **Lessons Learned**

- Planning database schema evolution from the start saves significant refactoring time
- Pure FP functions dramatically reduce debugging time for complex calculations
- Comprehensive testing pays dividends during feature additions and refactoring

### **Value-Added Features**

- Guest Mode: Quick testing capability with pre-loaded demo data
- Mood Tracking: Optional mood scores providing additional insights
- Soft Deletion: Data preservation while maintaining clean user interfaces
- Input Validation: Multi-level validation ensuring data integrity
- Visual Feedback: Emojis and colors enhancing user experience

## **PROJECT IMPACT**

### **Education Development**

This project successfully demonstrates:

- Advanced OOP principles (inheritance, polymorphism, encapsulation, abstraction)
- Functional programming concepts (pure functions, immutability, higher-order functions)
- Software design patterns in practice
- Production-ready Python development practices
- Comprehensive testing strategies

### **Professional Development**

The implementation showcases skills in:

- System architecture and design
- Database design and optimization
- Security best practices
- User experience design
- Documentation and collaboration

## GitHub Repository

Project Link: [https://github.com/ipadeolaoluwatoyin7880/habit\\_tracker](https://github.com/ipadeolaoluwatoyin7880/habit_tracker)

The repository contains:

- Complete source code with comprehensive documentation
- Installation and setup instructions
- Test suite with 40+ passing tests
- Sample data and demo configuration
- Phase 1 (Conception) and Phase 2 (Development) documents

## Conclusion

This habit tracker app exemplifies a successful blend of modern software engineering methods. The clear architecture, thorough testing, and thoughtful user experience design build a foundation that is both educational and professionally relevant. The project provides a solid foundation for future enhancements while delivering a reliable, user-friendly habit tracking experience.

The application not only satisfies all specified requirements but also surpasses them with thoughtful additional features and solid implementation. It demonstrates the effectiveness of combining object-oriented and functional programming paradigms in modern Python development.