



SDG BLOCKCHAIN ACCELERATOR

Prototype (PoC) Report – Template

1. Project Information

- **Project Name:** Sun4School
- **Challenge & UNDP Office:** UNDP Mauritius
- **Document Version:** Prototype (PoC) Report V1

2. Project Overview

(Briefly describe what the prototype does and its main features.)

This prototype implements a decentralized crowdfunding platform on Cardano for financing solar energy installations in Mauritian national schools. Users can browse school projects, invest or donate in fractional amounts (starting at MUR 100), and receive tokenized ownership shares (manually distributed in MVP). Key features include project listings with progress tracking, secure payment integration (simulated wallets), an impact dashboard for metrics (e.g., funds raised, CO₂ offsets), and future smart contracts for automated fund distribution and token airdrops, ensuring transparency and alignment with SDGs 4, 5, 7, and 13.

3. Repository Structure

(Outline how your code and related files are organized.)

Suggested Structure:

```
/project
  /cmd      → Entry points (app, migration, worker)
  /src      → Source code
    /apps   → Application logic, handlers, middleware
    /config → Configuration management
    /sql    → Database schemas and migrations by domain
  /tests    → Ginkgo/Gomega test files
  /docs     → API documentation and Swagger specs
  /bin      → Compiled binaries
  /tmp      → Temporary files
```

4. Build Instructions

(Explain how to compile/build the project.)

```
# Install Go dependencies
go get
# Build main API server
go build -o ./bin/app cmd/app/main.go
# Build background worker
go build -o ./bin/worker cmd/worker/main.go
# Build migration tool
go build -o ./bin/migrate cmd/migrate/main.go
```

5. Test Instructions

(Explain how to run unit and integration tests.)

```
# Run all tests using Go
go test ./tests/...
# Run with Ginkgo for BDD-style testing
ginkgo ./tests/
# Run specific test files
go test ./tests/auth_test.go
go test ./tests/projects_test.go
```

6. Deployment Instructions

(Explain how to deploy the project.)

```
# Setup configuration
cp .tmp.config.yml config.yml
# Start dependencies
sudo docker-compose up -d
# Run database migrations
go run cmd/migration/main.go up
# Start API server
```

```
go run cmd/app/main.go
# Start background workers (separate terminal)
go run cmd/worker/main.go
```

7. Testnet / Emulator Results

(Provide evidence of successful deployment and contract execution.)

N/A

8. Dependencies & Environment

- **Go:** 1.24.1
- **PostgreSQL:** 14.4-alpine (via Docker)
- **NATS:** Message queue service
- **Cardano Preprod:** Testnet for blockchain payments
- **External Services:** GoAccount, GoPay, Google Cloud Storage

9. Demo / Walkthrough

- Provide a short video, GIF, or screenshots demonstrating the prototype in action.
- Show minting, transferring, or retiring tokens as applicable.

10. Remaining Issues / Next Steps

(Mention pending improvements, planned features, or optimizations.)

- Complete Cardano blockchain integration testing
- Add comprehensive API documentation
- Implement monitoring and logging system
- Security audit for payment processing
- Performance optimization for database queries
- Add end-to-end testing for complete user workflows