# Stormwater Infrastructure Assessment Model

## Final Project Deliverables

This document contains all the requested deliverables for the project wrap-up and transition to the next capstone student team.

## 1. Project Manual

### 1.1 Project Overview

The Stormwater Infrastructure Assessment Model is a comprehensive web application designed to evaluate and analyze stormwater control measurement infrastructure. The application provides a detailed assessment framework based on industry standards with a 0-10 scoring system across multiple domains:

* Infrastructure Condition Assessment
* System Functionality Assessment
* Time-Effectiveness Assessment
* Cost-Effectiveness Assessment
* Environmental and Social Impact Assessment

### 1.2 System Architecture

The application uses the following technology stack:

* **Frontend**: Streamlit (Python-based web application framework)
* **Backend**: Python 3.11
* **Database**: SQLite (local database for data persistence)
* **Visualizations**: Plotly, Folium
* **Reporting**: ReportLab (PDF generation)

The architecture follows a modular design with the following key components:

1. **Views**: UI components for different sections of the application
2. **Components**: Reusable UI elements and visualization tools
3. **Utils**: Backend functionality including database handling, authentication, and report generation
4. **Data Models**: Database schema and data structures

### 1.3 Installation & Setup

To run the application locally:

1. Clone the repository from GitHub
2. Install the required dependencies:

* pip install -r requirements.txt

1. Run the application:

* streamlit run app.py

### 1.4 User Guide

#### Authentication

* The application supports user authentication with role-based access control
* Default admin credentials are provided in the admin passcode section below
* New users can be created through the admin panel

#### Project Management

1. Create or join a project from the projects page
2. Add team members to collaborate on assessments
3. View project metrics and historical assessment data

#### Assessment Process

1. Navigate to the Assessment page
2. Complete the assessment forms for each domain:
   * Condition Assessment (structural integrity, damage levels, etc.)
   * Functionality Assessment (hydraulic and hydrological performance)
   * Time-Effectiveness Assessment (lifespan, maintenance metrics)
   * Cost-Effectiveness Assessment (ROI, lifecycle costs)
   * Environmental and Social Impact Assessment
3. Add infrastructure points with geographical locations
4. Submit the assessment

#### Dashboard

The dashboard provides detailed visualizations: - Overall infrastructure score (0-10) - Domain-specific scores - Condition charts for individual assets - Performance radar charts - Environmental impact metrics - Geographical map of infrastructure points - Historical trends analysis

#### Reports

* Generate comprehensive PDF reports from the dashboard
* Reports include detailed metrics, scores, and city-specific recommendations
* Download and share reports with stakeholders

### 1.5 API Documentation

The application does not currently expose external APIs, but internal modules are documented as follows:

#### Database Module (utils/db.py)

* init\_database(): Initialize database schema
* get\_db(): Get database connection
* create\_project(name, description, created\_by): Create a new project
* add\_project\_member(project\_id, user\_id, role): Add member to project
* get\_user\_projects(user\_id): Retrieve projects for a user
* save\_assessment(data): Save assessment data
* get\_assessments(user\_id, project\_id): Retrieve assessments

#### Authentication Module (utils/auth.py)

* init\_auth(): Initialize authentication state
* check\_auth(): Handle user authentication
* create\_user(username, password, is\_admin): Create a new user

#### Reporting Module (utils/report.py)

* generate\_report(assessment\_data, project\_name): Generate PDF report

## 2. Final Presentation Slides

The presentation slides are available in the repository under /doc\_assets/final\_presentation.pptx and cover the following topics:

1. **Project Overview**: Introduction to stormwater infrastructure assessment challenges
2. **Solution Architecture**: Technical architecture and components
3. **Assessment Model**: Detailed breakdown of the 0-10 scoring methodology
4. **Data Visualization**: Dashboard and reporting capabilities
5. **Future Directions**: AI integration opportunities
6. **Demo**: Live demonstration of key features
7. **Q&A**: Frequently asked questions

## 3. Online Database Access

The current implementation uses SQLite for data storage, which is a local file-based database. For production deployment, we recommend migrating to a cloud-based database solution such as:

* MongoDB Atlas (NoSQL)
* PostgreSQL (relational database)

The database schema is as follows:

**Users Table** - id (primary key) - username - password\_hash - is\_admin - created\_at

**Projects Table** - id (primary key) - name - description - created\_by (user\_id) - created\_at

**Project\_Members Table** - id (primary key) - project\_id (foreign key) - user\_id (foreign key) - role - joined\_at

**Assessments Table** - id (primary key) - user\_id (foreign key) - project\_id (foreign key) - data (JSON) - timestamp

## 4. Website Password

The website is secured with user authentication. The default accounts are:

**Regular User** - Username: user - Password: password123

These credentials provide access to basic application functionality including creating projects, performing assessments, and generating reports.

## 5. Admin Passcode

The admin account has additional privileges for user management and system configuration:

**Admin Account** - Username: admin - Password: admin123

Admin features include: - User management (create/edit/delete users) - View system statistics - Access activity timeline - Configure system settings

## 6. GitHub Repository Link

The complete source code is available at:

<https://github.com/stormwater-assessment-model/infrastructure-assessment>

Repository structure:

/  
├── .streamlit/ # Streamlit configuration files  
├── app/ # Application modules  
├── attached\_assets/ # Sample documents and resources  
├── components/ # UI components and visualizations  
│ ├── charts.py # Data visualization components  
│ ├── forms.py # Assessment forms  
│ ├── gis\_integration.py # GIS functionality  
│ ├── heat\_map.py # Heat map visualization  
│ ├── map\_view.py # Map integration  
│ ├── org\_tree.py # Organization structure  
│ └── sunburst.py # Sunburst chart visualization  
├── doc\_assets/ # Documentation resources  
├── temp/ # Temporary files  
├── utils/ # Utility functions  
│ ├── auth.py # Authentication handling  
│ ├── db.py # Database operations  
│ ├── documentation\_generator.py # Generates user manual  
│ └── report.py # PDF report generation  
├── views/ # Application views  
│ ├── admin.py # Admin panel  
│ ├── assessment.py # Assessment forms  
│ ├── dashboard.py # Main dashboard  
│ ├── documentation.py # Documentation page  
│ ├── home.py # Home page  
│ └── projects.py # Project management  
├── app.py # Main application entry point  
├── pyproject.toml # Project dependencies  
└── README.md # Project documentation

## 7. Next Steps for AI Integration

For the next capstone team focusing on AI integration, we recommend:

1. **Data Collection & Preprocessing**:
   * Develop data pipelines to collect historical assessment data
   * Create preprocessing utilities for standardizing input formats
   * Implement data augmentation for limited datasets
2. **Model Development**:
   * Develop predictive models for infrastructure degradation
   * Create recommender systems for maintenance prioritization
   * Implement anomaly detection for identifying outlier conditions
3. **Integration Points**:
   * Integrate AI predictions into the dashboard
   * Add AI-powered recommendations in reports
   * Create feedback loops for model improvement
4. **Deployment Considerations**:
   * Model versioning and management
   * Performance monitoring
   * Transfer learning capabilities

The current assessment data structure is already compatible with AI applications, with well-defined numerical metrics and categorical variables suitable for machine learning models.

## 8. Contact Information

For any questions or assistance with the project, please contact:

* Project Manager: [project.manager@university.edu](mailto:project.manager@university.edu)
* Technical Lead: [tech.lead@university.edu](mailto:tech.lead@university.edu)
* Department Chair: [department.chair@university.edu](mailto:department.chair@university.edu)

This project was developed as part of the University Engineering Capstone Program, 2025.