Islamabad City Management System

A Comprehensive C++ City Infrastructure Management Application

---

Project Overview

The **Islamabad City Management System** is a console-based application (with optional SFML graphical visualizations) that simulates the management of various city infrastructure components. The system demonstrates comprehensive use of data structures and algorithms including:

- **Graphs** (adjacency lists)
- **Hash Tables** (separate chaining)
- **Heaps** (min-heap for Dijkstra, max-heap for priority systems)
- **Trees** (N-ary trees, B-trees)
- **Queues** (circular queues)
- **Stacks** (array-based)
- **Linked Lists** (singly linked)
- **Dijkstra's Algorithm** for shortest path

Modules:

1. **Transport** - Bus stops, routes, pathfinding, school buses
2. **Medical** - Hospitals, pharmacies, medicines, emergency beds
3. **Education** - Schools, departments, teachers, students
4. **Commercial** - Malls, products, inventory, delivery routes
5. **Facilities** - Public facilities (parks, libraries, etc.)
6. **Population** - Citizen records, B-tree indexing
7. **Bonus Modules** - Airport/Railway integration
8. **SFML Visualizations** - Graphical city map, network graphs, statistics

---

Installation & Setup

Step 1: Install C++ Compiler

Windows:

1. Download **MinGW-w64** from: https://sourceforge.net/projects/mingw-w64/
2. Run installer, select architecture (x86_64 for 64-bit)
3. Add MinGW `bin` folder to system PATH:
   - Right-click "This PC" → Properties → Advanced System Settings
   - Environment Variables → System Variables → Path → Edit
   - Add: `C:\mingw64\bin`

   Verify installation:

   ```
    g++ --version
   ```
4.

Linux (Ubuntu/Debian):

```
sudo apt update

sudo apt install build-essential g++

g++ --version
```

macOS:

```
xcode-select --install

g++ --version
```

Step 2: Install SFML (Optional - for visualizations)

Windows:

1. Download SFML from: https://www.sfml-dev.org/download.php
2. Extract to `C:\SFML`
3. Copy DLL files from `C:\SFML\bin` to project directory
4. Required DLLs:
   - `sfml-graphics-2.dll`
   - `sfml-window-2.dll`

- `sfml-system-2.dll`

Linux:

sudo apt install libsfml-dev

Clone or Download Project

# Option 1: Clone with Git

git clone <repository-url>

cd islamabad-city-management

# Option 2: Download ZIP

# Extract the ZIP file to a folder

Organize Project Files

Ensure your project directory looks like this:

```
islamabad-city-management/
│
├── Source.cpp              # Main program
├── Utils.h                 # Utility functions
├── GlobalLocationManager.h    # Location tracking
├── CityGraph.h             # Unified city graph
├── Transport.h             # Transport module
├── Medical.h               # Medical module
├── Education.h              # Education module
├── Commercial.h             # Commercial module
├── Facilities.h             # Facilities module
├── Population.h             # Population module
├── BonusModules.h           # Bonus features
├── Sfmlvisualizer.h         # SFML graphics
│
├── arial.ttf                # Font file (required for SFML)
│
├── Data Files (CSV):
│   ├── stops.csv
│   ├── busstops.csv
│   ├── roads.csv
│   ├── buses.csv
│   ├── hospitals.csv
│   ├── pharmacies.csv
│   ├── schools.csv
│   ├── malls.csv
│   ├── products.csv
│   ├── facilities.csv
│   └── citizens.csv
│
└── README.md
```

Running the Application

Method 1: Compile Without SFML (Console Only)

If you don't need graphical visualizations:

```
# Compile (standard C++)

g++ -o city_management Source.cpp -std=c++11


# Run

./city_management          # Linux/macOS

city_management.exe        # Windows
```

Method 2: Compile With SFML (Full Features)

Windows (MinGW):

```
g++ -o city_management Source.cpp ^

    -std=c++11 ^

    -IC:\SFML\include ^

    -LC:\SFML\lib ^

    -lsfml-graphics -lsfml-window -lsfml-system ^

    -DSFML_STATIC
```

Note: Replace `C:\SFML` with your actual SFML installation path

Linux:

```
g++ -o city_management Source.cpp \
```

-std=c++11 \

-lsfml-graphics -lsfml-window -lsfml-system

macOS:

g++ -o city_management Source.cpp \

-std=c++11 \

-I/usr/local/include \

-L/usr/local/lib \

-lsfml-graphics -lsfml-window -lsfml-system

Method 3: Using Visual Studio (Windows)

1. Open **Visual Studio 2019/2022**
2. Create new **Empty C++ Project**
3. Add all `.h` files and `Source.cpp` to project
4. Configure SFML (if using):
   - **Project Properties → C/C++ → General → Additional Include Directories**
     - Add: `C:\SFML\include`
   - **Project Properties → Linker → General → Additional Library Directories**
     - Add: `C:\SFML\lib`
   - **Project Properties → Linker → Input → Additional Dependencies**
     - Add: `sfml-graphics.lib;sfml-window.lib;sfml-system.lib;`
5. Build (Ctrl+Shift+B)
6. Copy SFML DLLs to output directory
7. Copy `arial.ttf` to output directory
8. Run (Ctrl+F5)

Method 4: Using Code::Blocks

1. **File → New → Project → Console Application**
2. Add all header files and Source.cpp

3. **Build → Build Options → Linker Settings**
4. Add SFML libraries:
   - `sfml-graphics`
   - `sfml-window`
   - `sfml-system`
5. Build and Run (F9)

Project Structure

Header Files Overview

File	Purpose	Key Data Structures
`Utils.h`	Utility functions (parsing, hashing, I/O)	String manipulation, hash functions
`GlobalLocationManager.h`	Prevents location conflicts	Linked list
`CityGraph.h`	Unified city-wide graph	Adjacency list graph
`Transport.h`	Bus system management	Graph, hash table, circular queue, stack, min-heap
`Medical.h`	Healthcare facilities	Dynamic hash table, max-heap
`Education.h`	School system	N-ary tree, hash table, max-heap

`Commercial.h`	Shopping malls	Graph, hash table
`Facilities.h`	Public facilities	Graph, hash table by type
`Population.h`	Citizen records	B-tree, 4-level tree
`BonusModules.h`	Airport/Railway hubs	Linked lists, schedules
`Sfmlvisualizer.h`	Graphical visualizations	SFML rendering

Features

Core Features:

1. Transport Management

- Add/update/delete bus stops
- Create road networks (graph)
- Register buses with routes
- Find shortest paths (Dijkstra's algorithm)
- Passenger queue management (circular queue)
- Route history tracking (stack)
- **School bus system** with route simulation

2. Medical Services

- Register hospitals and pharmacies
- Add doctors and medicines
- **Dynamic hash table** with automatic resizing
- Emergency bed availability (max-heap)
- Find nearest hospital

- Medicine search

3. Education System

- **N-ary tree** hierarchy (School → Department → Section → Student)
- Add teachers and students
- School rankings (max-heap)
- Subject-based search (hash table)
- Transfer students between sections
- Display organograms

4. Commercial (Malls)

- Register malls and products
- Inventory management
- Product search by name/category
- Find nearest mall with product
- Shortest delivery path (Dijkstra)

5. Public Facilities

- Register facilities (parks, libraries, gyms, etc.)
- Amenity management
- Rating system
- Find nearest facility by type
- Shortest path between facilities

6. Population Management

- **B-tree indexing** by CNIC
- 4-level hierarchy (Sector → Street → House → Citizen)
- Search by CNIC (O(log n))
- Generate reports (occupation, age distribution)
- Transfer citizens

7. Bonus: Airport/Railway Integration

- Register transport hubs (airports, railway stations)
- Manage schedules (departures/arrivals)
- Update flight/train status
- Connect hubs to bus network

8. SFML Visualizations

- **City Map**: View all locations on a map
- **Graph Network**: See road/connection networks

- **Bus Route Viewer**: Animated bus movement
- **Population Heatmap**: Bar chart by sector
- **Statistics Dashboard**: Real-time city stats

Quick Start Guide

First-time users:

Compile (without SFML for simplicity):

```
g++ -o city_management Source.cpp -std=c++11
```

1.

Run:

```
./city_management
```

2.
3. **Initial Setup**:
   - When prompted "Load data from files?", type `yes`
   - If CSV files exist, data will load automatically
   - If not, select option `8` from main menu to manually load
4. **Try These Features**:
   - **Main Menu → 1 (Transport)**: Add a bus stop, register a bus
   - **Main Menu → 2 (Medical)**: Register a hospital, add medicines
   - **Main Menu → 7 (Statistics)**: View system-wide stats
   - **Main Menu → 10 (SFML Visualizations)**: See graphical views (if SFML installed)
5. **Explore Algorithms**:
   - **Transport Menu → 22**: Find shortest path (Dijkstra's algorithm)
   - **Medical Menu → 8**: View bed availability (max-heap)
   - **Education Menu → 7**: Show school rankings (max-heap)

License & Credits

Project: Islamabad City Management System
Course: Data Structures & Algorithms
Language: C++11
External Libraries: SFML (optional, for visualizations)

Data Structures Implemented:

- Graphs (Adjacency List)
- Hash Tables (Separate Chaining, Dynamic Resizing)
- Heaps (Min-Heap, Max-Heap, Dynamic Array-Based)
- Trees (N-ary Tree, B-Tree)
- Queues (Circular Queue)
- Stacks (Array-Based)
- Linked Lists (Singly Linked)

Algorithms Implemented:

- Dijkstra's Shortest Path Algorithm
- Heap Sort (via heap operations)
- Hash Functions (Polynomial Rolling, Sum Hash)
- Tree Traversals (DFS, Pre/Post-order)
- B-Tree Operations (Insert, Search, Split)

Educational Value

This project demonstrates:

1. **Real-world application** of data structures
2. **Algorithm complexity analysis** (O(1), O(log n), O(n), O((V+E) log V))
3. **Trade-offs** in data structure selection
4. **Memory management** in C++
5. **Modular design** and code organization
6. **File I/O** and data persistence
7. **Graph algorithms** for practical problems
8. **Dynamic memory** and resizing strategies

Conclusion

You now have everything needed to:

- Compile and run the application
- Load data from CSV files
- Use all modules and features
- Enable SFML visualizations (optional)
- Troubleshoot common issues
- Understand the codebase

Next Steps:

1. Run the program and explore features
2. Read the detailed report (REPORT.md) for algorithm analysis
3. Try modifying code to add new features
4. Test with your own CSV data

Enjoy managing your virtual city!

Made By:
M.Taha (i240584)
Ali Bazmi (i240623)
M.Hassaan (i240717)