Sprint 1 Quick Reference Checklist

⚡ Quick Start (30 minutes)

```bash
# 1. Setup environment
python -m venv resq_env
source resq_env/bin/activate  # or: resq_env\Scripts\activate on Windows

# 2. Install dependencies
pip install osmnx networkx numpy matplotlib jupyter pandas
pip freeze > requirements.txt

# 3. Create project structure
mkdir -p resq-graph/{src,tests,data,docs,notebooks}
cd resq-graph
touch src/__init__.py tests/__init__.py

# 4. Start Jupyter
jupyter notebook
```

📋 US-001: Initialize Development Environment

Goal

✅ All packages installed and imports verified

Checklist

- □ Virtual environment created
- □ `requirements.txt` generated
- □ All imports tested (osmnx, networkx, numpy, matplotlib)
- □ Folder structure created (`src/`, `tests/`, `data/`, `docs/`)
- □ `.gitignore` created

Quick Test

```python
```

```
import osmnx, networkx, numpy, matplotlib
print("✓ All good!")
```

Time: 30-45 minutes

🗺️ US-002: Select and Validate Sandbox Map

Goal

✅ Model Town, Lahore boundaries documented and OSM data quality verified

Checklist

□ Bounding box coordinates defined:

- North: 31.5350
- South: 31.5050
- East: 74.3700
- West: 74.3470

□ OSM data downloaded for Model Town

□ Graph stats computed (nodes, edges, connectivity)

□ No major isolated nodes

□ `data/model_town_metadata.txt` created with documentation

Key Command

```python
import osmnx as ox
G = ox.graph_from_bbox(
    north=31.5350, south=31.5050, east=74.3700, west=74.3470,
    simplify=True, network_type='drive'
)
print(f"Nodes: {G.number_of_nodes()}, Edges: {G.number_of_edges()}")
```

Expected Output

```
Nodes: ~2000-3000
Edges: ~4000-6000
Is weakly connected: True
Isolated nodes: 0 or very few
```

Time: 1-1.5 hours

🔧 US-003: Implement Map Baking Pipeline

Goal

✅ Raw OSM data converted to clean graph and saved as GraphML

Checklist

- □ Function: `remove_isolated_nodes(G)` - removes degree-0 nodes
- □ Function: `get_graph_stats(G)` - computes graph statistics
- □ Function: `bake_map()` - main pipeline
- □ File saved: `data/modeltown.graphml`
- □ File saved: `data/modeltown_stats.json`
- □ Graph reloaded and verified

Key Code

```python
# Remove isolated nodes
isolated = [node for node in G.nodes() if G.degree(node) == 0]
G_clean = G.copy()
G_clean.remove_nodes_from(isolated)

# Save
import networkx as nx
nx.write_graphml(G_clean, "data/modeltown.graphml")

# Verify
G_test = nx.read_graphml("data/modeltown.graphml")
```

Expected Files

```
data/
├── modeltown.graphml      (Graph file, ~1-5 MB)
├── modeltown_stats.json    (Statistics)
└── model_town_metadata.txt  (Documentation)
```

Time: 1.5-2 hours

📊 Sprint 1 Summary

Deliverables

Code Files

```
src/
├── __init__.py
└── map_loader.py          ← Main module (or in notebook)

tests/
└── test_map_loader.py     ← Unit tests

notebooks/
├── 01_environment_setup.ipynb
├── 02_map_validation.ipynb
└── 03_map_baking.ipynb
```

Data Files

```
data/
├── modeltown.graphml
├── modeltown_stats.json
└── model_town_metadata.txt
```

Documentation

```
├── README.md
├── requirements.txt
└── .gitignore
```

Success Criteria

□ All US-001 tasks complete

□ All US-002 tasks complete

□ All US-003 tasks complete

□ Code follows PEP 8 style

□ Basic tests pass

□ Documentation present

□ All files in version control (git)

Time Estimate

Total: 8-12 hours (light sprint)

🎯 Key Metrics to Track

After Sprint 1, you should have:

Metric	Value
Graph Nodes	~2000-3000
Graph Edges	~4000-6000
Weakly Connected	✓ Yes
Isolated Nodes Removed	N
GraphML File Size	~1-5 MB
Setup Time	~30 min
Testing Time	~30 min

🚀 Ready for Sprint 2?

After Sprint 1 completion, you're ready for Sprint 2 which will:

- Load `modeltown.graphml`
- Implement A* pathfinding algorithm
- Create path visualizations
- Write unit tests for pathfinding

💡 Pro Tips

1. Save Progress Often

python

```python
import pickle
with open('data/graph_backup.pkl', 'wb') as f:
    pickle.dump(G_clean, f)
```

2. Use Git from Day 1

bash

```bash
git init
git add .
git commit -m "Sprint 1: Initial environment setup"
```

3. Document Assumptions

python

```python
# Comment your code!
# Model Town coordinates from Google Maps
# Bounding box: ~15km x 15km area
north, south, east, west = 31.5350, 31.5050, 74.3700, 74.3470
```

4. Keep Notebooks Organized

- One notebook = one sprint task (or related task)
- Clear cell structure (markdown headers)
- Document purpose at top

5. Test Incrementally

python

```python
# Don't wait until end to test!
assert G.number_of_nodes() > 100
assert nx.is_weakly_connected(G_clean)
print("✓ Assertions passed")
```

🆘 Common Issues & Fixes

"OSM download timeout"

```python
import osmnx as ox
ox.config(timeout=180)
```

"ImportError: No module named 'osmnx'"

```bash
pip install --upgrade osmnx
```

"Graph is disconnected"

→ This is OK! Just document it
→ May indicate OSM data quality issue

"GraphML file not found"

→ Check file path is correct → Make sure `data/` directory exists

"Jupyter notebook not starting"

```bash
jupyter --version  # Check installation
jupyter notebook --no-browser
```

📞 When to Move to Sprint 2

You're ready when:

- ✅ All imports work (`test_imports.py` passes)
- ✅ Graph downloaded and validated (stats logged)
- ✅ Graph cleaned (isolated nodes removed)
- ✅ Graph saved as GraphML
- ✅ Graph can be reloaded from file
- ✅ Tests written and passing
- ✅ Documentation complete

📈 Expected Time Breakdown

Task	Time	Start	End
US-001: Environment	0.5-1 hr	Day 1	Day 1
US-002: Validation	1-1.5 hrs	Day 1-2	Day 2
US-003: Baking	1.5-2 hrs	Day 2-3	Day 3
Testing & Docs	1-2 hrs	Day 3-4	Day 4
Total	**8-12 hrs**		**~1 week**

📚 Resources

- **OSMnx Docs:** https://osmnx.readthedocs.io/
- **NetworkX Docs:** https://networkx.org/
- **Jupyter Docs:** https://jupyter.org/
- **PEP 8 Style:** https://pep8.org/
- **Git Guide:** https://git-scm.com/doc

✅ Final Checklist Before Moving to Sprint 2

□ Virtual environment working

□ All packages installed and tested

□ Project structure created

□ Model Town map downloaded and validated

□ Graph cleaned and saved as GraphML

□ Statistics documented

□ Code follows PEP 8

□ Basic tests written and passing

□ README.md updated

□ Files in version control (git)

- □ Ready to load graph and implement A*

_You're all set! 🎉 Ready for Sprint 2: A Pathfinding_*