SIMULATION TOOL USING GEOGRAPHIC INFORMATION SYSTEM WITH DIGITAL TWIN TECHNOLOGY INTEGRATION

Madhuwantha G K O

(IT21802058)

BSc (Hons) Degree in Information Technology Specialized in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

August 2025

SIMULATION TOOL USING GEOGRAPHIC INFORMATION SYSTEM WITH DIGITAL TWIN TECHNOLOGY INTEGRATION

Madhuwantha G K O

IT21802058

BSc (Hons) Degree in Information Technology Specialized in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology Sri Lanka

August 2025

DECLARATION

I declare that this report entitled *“Virtual Modeling for Urban Heat Island (UHI) Prediction and Simulation Tool Development”* is my own work and does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other institution of higher learning. To the best of my knowledge, this report does not contain any material previously published or written by another person except where due reference is made in the text.

Name	Student ID	Signature
Madhuwantha G K O	IT21802058	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

29/08/2025

-- --

Signature of the Supervisor: Date:

ABSTRACT

The Urban Heat Island (UHI) effect, characterized by elevated temperatures in urban areas compared to surrounding rural regions, poses significant challenges to environmental sustainability, energy efficiency, and public health. With increasing urbanization, accurate modeling and prediction of UHI have become essential for urban planning and climate mitigation strategies.

This research presents the development of a **simulation tool** that integrates **Geographic Information System (GIS) data, 3D modeling (Digital Twin Technology), environmental metadata, weather API data, and MATLAB-based thermal simulations** to predict UHI effects at building and city-block scales. A **digital twin** of supposed building is created using Blender and React-Three.js, incorporating GIS datasets (roads, buildings, vegetation, and water bodies). Metadata such as material properties, wind speed, surrounding temperature, humidity, and sun exposure are embedded into the extracted csv file from model to enable physics-based simulations.

The tool computes **sunlight exposure** using SunCalc, **environmental conditions** (temperature, humidity, wind) using real-time weather APIs, and compiles this data into a **CSV pipeline** for MATLAB Simulink simulations. The Simscape Thermal library is employed to model heat transfer and temperature distribution across building components (e.g., walls, roofs). The simulation outputs provide spatial temperature profiles (e.g., *Roof: 45°C, Wall: 37°C*), enabling UHI prediction and comparative analysis under different weather conditions and material properties.

Experimental validation demonstrates the feasibility of integrating **GIS-driven 3D digital twins with MATLAB-based physics simulations** for urban climate analysis. The findings contribute to advancing digital twin–based urban sustainability planning, offering potential applications in smart cities, green infrastructure design, and environmental policy.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who supported and contributed to the successful completion of this research project.

First and foremost, I extend my deepest appreciation to my supervisor, Mr. Vishan Jayasinghearachchi,, co- supervisor, Ms. Kaushalya Rajapakse, and external supervisor, Dr. Rajitha De Silva for their invaluable guidance, encouragement, and expertise throughout the research journey. Their constructive feedback and constant support were instrumental in shaping the direction and quality of this work.

I am especially thankful to the academic staff of the Department of Software Engineering at the Sri Lanka Institute of Information Technology for providing the necessary resources and a stimulating environment that fostered research and innovation.

I would also like to acknowledge my project group members for their collaboration and dedication during the stages of the research proposal. Furthermore,

I am grateful to the participants who generously contributed their time during the data collection and evaluation phases. Your participation was vital in validating the research outcomes.

To my colleagues and friends, thank you for your continuous encouragement, valuable discussions, and moral support, which greatly enriched my research experience.

Finally, I am deeply indebted to my family for their unwavering support, patience, and motivation throughout this journey. Their belief in me provided the strength to overcome challenges and complete this work successfully.

This research stands as a testament to the collective efforts, mentorship, and inspiration provided by all those involved, and I remain sincerely grateful for their contributions

Table of Contents

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CSV	Comma-Separated Values
CFD	Computational Fluid Dynamics
ENVI-met	Environmental Microclimate Simulation Software
GIS	Geographic Information System
GLTF/GLB	Graphics Language Transmission Format (binary format for 3D models)
IoT	Internet of Things
MATLAB	Matrix Laboratory (Numerical Computing Environment)
ML	Machine Learning
OSM	OpenStreetMap
PV	Photovoltaic
SDLC	Software Development Life Cycle
STL	Stereolithography (3D model file format)
SunCalc	Sun Position and Exposure Calculation Library
UHI	Urban Heat Island
UI	User Interface
3D	Three Dimensional

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1 INTRODUCTION

1.1 Background

Urbanization is accelerating globally, bringing both opportunities and challenges. While modern cities drive economic development and innovation, they also face environmental threats such as air pollution, increased energy consumption, and elevated temperatures in built-up regions. One of the most significant environmental challenges is the **Urban Heat Island (UHI) effect**, which refers to the temperature difference between densely built urban areas and their surrounding rural regions. This occurs because urban surfaces such as concrete, asphalt, and rooftops absorb and retain more heat compared to vegetation-covered areas. The absence of green cover and water bodies further reduces natural cooling mechanisms. As a result, urban residents experience hotter conditions, increased energy demand for cooling, and greater risks of heat-related health issues.

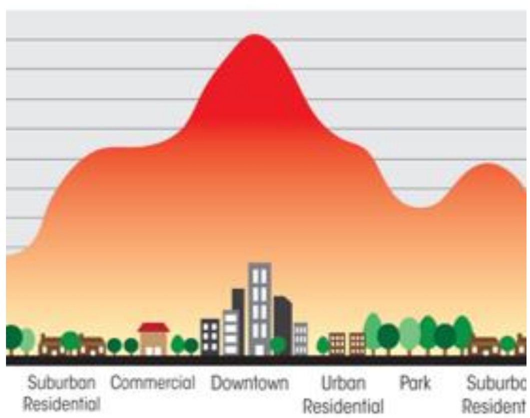


Figure 1-1 Illustration of UHI effect – urban vs. rural temperature profile.

The UHI phenomenon highlights the urgent need for **urban simulation tools** that can analyze the impact of different environmental and architectural factors. Such tools allow city planners and engineers to evaluate design choices and introduce interventions such as reflective materials, green roofs, or optimized building layouts.

1.2 Problem Context

Traditional approaches to analyzing UHI rely on field measurements and large-scale remote sensing. While these methods provide valuable insights, they are often limited in resolution and adaptability. For example, remote sensing can detect temperature variations at the city scale but may fail to capture micro-level interactions such as the effect of a specific roof material or the orientation of a single building.

Existing computational tools also tend to be specialized and isolated. Some platforms focus on microclimate simulations, while others emphasize energy efficiency in buildings. However, there is a **lack of integrated solutions** that combine:

- **Geographic Information System (GIS) data** for accurate city layouts,
- **3D modeling of urban environments**,
- **Real-time environmental data** such as temperature, humidity, and wind speed, and
- **Physics-based simulations** for thermal behavior of materials.

Without such integration, urban planners are left with fragmented analyses that cannot fully capture the complexity of UHI.

1.3 Motivation

The motivation for this research arises from the urgent need to develop **predictive tools for sustainable urban planning**. With rising global temperatures and climate change, urban areas in tropical countries face intensified heat conditions. In such environments, the UHI effect not only increases discomfort but also leads to higher energy bills, heat stress, and long-term health risks. Urban planning authorities, architects, and environmental engineers require **decision-support systems** that can simulate different scenarios before implementation. For instance:

- How would plant more trees reduce average street temperatures?
- What impact would reflective roofing have on building-level heat retention?
- How would changes in wind flow due to new building arrangements affect UHI intensity?

This project is motivated by the vision of addressing these questions through an integrated **digital twin simulation tool**.

1.4 Research Problem

Despite significant progress in computational modeling and climate studies, the following challenges remain unaddressed:

- **Fragmentation of tools**: No single framework integrates GIS, real-time weather data, 3D urban models, and simulation engines.
- **Limited focus on building components**: Current models rarely provide insights at the scale of individual walls, roofs, and streets.
- **Static analysis**: Many tools do not adapt dynamically to changing environmental conditions such as wind, humidity, and solar radiation.
- **Lack of decision-making support**: Simulation results are often too technical or abstract for urban planners to translate into practical policy.

Research Problem Statement:
How can an integrated digital twin simulation tool be developed to combine GIS datasets, 3D modeling, environmental data, and MATLAB-based physics simulations to predict and analyses Urban Heat Island?

1.5 Research Scope

The scope of this study is defined to ensure feasibility while maintaining practical relevance:

- **Geospatial Scope**: Focus on a selected urban area, modeled with GIS data for roads, buildings, vegetation, and water bodies.
- **Technical Scope**: Development of a simulation tool integrating 3D modeling, weather APIs, SunCalc sunlight exposure calculations, and MATLAB thermal simulations.
- **Output Scope**: Generation of localized temperature predictions, including component-level heat distribution (e.g., walls, roofs, pavements).
- **Exclusions**: The project does not focus on large-scale regional climate forecasting but instead emphasizes **building-level and neighborhood-level UHI modeling**.

1.6 Research Contribution

This study contributes in several ways:

1. **Novel Pipeline** – A fully integrated simulation tool combining GIS, 3D modeling, real-time data, and MATLAB simulations.
2. **Building-Level Analysis** – Ability to predict thermal behavior of individual building components.
3. **Decision Support** – Usable insights for architects, city planners, and policymakers to test different interventions.
4. **Advancement of Digital Twin Applications** – Demonstrates how digital twins can be applied in urban sustainability research.
5. **Practical Impact** – Supports smart city initiatives and climate adaptation strategies.

2 LITERATURE REVIEW

Urban Heat Island (UHI) research has grown significantly over the past decades due to its direct implications for climate resilience, urban sustainability, and public health. As cities expand and undergo rapid transformation, researchers have increasingly turned to diverse approaches such as **remote sensing, geographic information systems (GIS), 3D digital twin models, and computational simulation tools** to study and mitigate UHI. More recently, **machine learning and artificial intelligence (AI)** have emerged as complementary methods for predictive modeling. This chapter reviews the evolution of UHI research across these domains, highlighting their strengths, limitations, and relevance to the development of integrated simulation pipelines.

The **UHI phenomenon** has been well documented in both global and regional contexts. Studies consistently show that dense urbanization, replacement of natural land cover with impervious materials, and human-induced heat emissions led to elevated urban temperatures compared to surrounding rural zones. Lee et al. (2020) conducted a trend analysis of UHI intensity across Asian mega-cities, demonstrating that the magnitude of UHI strongly correlates with rapid land use change and urban sprawl [1]. Similarly, Kim (2023) emphasized the importance of integrating both human and environmental dimensions in understanding UHI, particularly in South Asian cities where rising temperatures disproportionately affect vulnerable communities [2]. While early research largely focused on documenting the existence of UHI, more recent studies seek to **quantify UHI intensity, identify its spatial variability, and evaluate mitigation strategies**. Ahmed et al. (2024) provided a comprehensive review of predictive approaches, concluding that conventional statistical techniques have now been surpassed by hybrid methods that incorporate AI models to improve accuracy [3].

One of the earliest and most widely used techniques in UHI research is **GIS and remote sensing**. Satellite imagery, such as Landsat 8 and Sentinel 2 data, has enabled high-resolution monitoring of land surface temperature (LST) at city-wide and regional scales. Thong et al. (2024) demonstrated the effectiveness of combining multiple satellite datasets to analyze UHI patterns in Ho Chi Minh City, revealing both seasonal and spatial differences in heat intensity [4]. GIS has also been applied to overlay urban features such as vegetation cover, road density, and building footprints with temperature maps, thereby identifying correlations between land use and UHI. Comparative analyses of tools such as ArcGIS and QGIS have highlighted the importance of computational efficiency in handling large-scale urban data transformations (Dębicka, 2018) [5]. However, while GIS and remote sensing are indispensable for identifying macro-level UHI patterns, they often lack the granularity to simulate **building-level thermal interactions**, limiting their role in micro-urban analysis.

To bridge this gap, researchers have increasingly explored **3D digital twin and simulation models** as tools for urban climate analysis. Digital twins enable virtual replication of real-world urban environments, embedding data such as geometry, material properties, and environmental conditions. They provide dynamic simulation environments that can test "what-if" scenarios, such as altering building orientation, adding vegetation, or applying reflective surfaces. Ahuja et al. (2011) introduced one of the earliest frameworks for environmental modeling integration, stressing the importance of interoperability between modeling platforms [6]. More recently, Pillai et al. (2025) reviewed digital twin applications in smart cities, highlighting their potential in UHI mitigation through advanced visualization and predictive modeling [7]. Visualization platforms such as React-Three.js have also expanded the accessibility of digital twins, enabling real-time 3D interaction with urban models in web applications (Beshai, 2020) [8].

Alongside visualization, computational fluid dynamics (CFD) and microclimate models have played an important role in analyzing thermal and wind flow conditions in complex urban terrains. Xue et al. (2025) demonstrated that CFD simulations can capture the interplay of wind dynamics and surface temperatures in densely built environments [9]. Similarly, ENVI-met, a microclimate modeling tool, has been widely validated against field data, particularly in studies assessing façade greening and vegetation impacts on thermal comfort (Voelker et al., 2022) [10]. However, while

CFD and ENVI-met provide strong insights into environmental processes, they are often computationally intensive and not easily integrated into flexible pipelines.

For building-level and material-specific thermal studies, **MATLAB and Simscape Thermal** have emerged as powerful platforms. EnergyPlus, developed by the U.S. Department of Energy, is commonly used for building energy modeling [11], but MATLAB Simscape offers an alternative that emphasizes modularity and integration with engineering workflows. MathWorks (2022) outlines how Simscape Thermal can model conduction, convection, and radiation processes for building components, with parameterized inputs for material conductivity, specific heat capacity, and thickness [12]. Such models are particularly valuable for simulating the thermal behavior of walls, roofs, and other building elements under dynamic environmental conditions. Recent studies have demonstrated that coupling MATLAB-based models with digital twin environments enables both accuracy and flexibility, allowing researchers to test multiple design interventions quickly.

In parallel with deterministic approaches, **machine learning and AI** are increasingly applied to UHI prediction. Ahmed et al. (2024) reviewed the use of artificial intelligence in UHI studies, noting that algorithms such as artificial neural networks (ANNs), support vector machines (SVMs), and random forests have achieved strong predictive accuracy when trained on remote sensing and meteorological data [3]. Ghorbany et al. (2024) similarly highlighted the role of machine learning in advancing UHI research, particularly in analyzing nonlinear interactions between land cover, climate variables, and anthropogenic activities [13]. These AI-driven models are particularly suited for large-scale prediction and trend analysis, though they often require significant datasets and may lack interpretability compared to physics-based simulations.

Recent research further underscores the need for **hybrid approaches** that combine physics-based models with AI-driven predictions. For example, sustainability-oriented studies have proposed combining thermal simulations with weather API data to create adaptive frameworks capable of predicting real-time UHI variations (WeatherAPI, 2025) [14]. Mitigation strategies such as urban greenery, reflective roofing, and shading interventions have also been tested through simulation–AI combinations, demonstrating measurable reductions in heat intensity (Irfeey et al., 2023) [15]. Novel applications such as solar exposure monitoring (Kluson et al., 2024) [16] provide further opportunities to enrich UHI studies by incorporating personal exposure data into broader urban models.

Collectively, the literature demonstrates that UHI research has evolved from descriptive documentation of heat intensity to **sophisticated simulation and prediction systems**. GIS and remote sensing remain vital for large-scale monitoring, while digital twins and MATLAB/Simscape Thermal provide detailed building-level analysis. Meanwhile, AI and machine learning methods offer predictive power that complement physics-based simulations. However, significant gaps remain in achieving **end-to-end integration** of these methods into user-friendly platforms. Existing studies tend to emphasize either macro-scale or micro-scale approaches but rarely combine them into cohesive systems. This gap highlights the importance of research focused on developing **integrated pipelines that merge GIS, 3D digital twins, weather APIs, and thermal simulation models**, ultimately enabling more practical and actionable UHI mitigation strategies for smart cities.

3 RESEARCH GAP & PROBLEM STATEMENT

3.1 Research Gap

The literature reviewed demonstrates significant progress in the study of the Urban Heat Island (UHI) effect, with contributions from disciplines such as climatology, urban planning, computer science, and environmental engineering. However, despite this progress, **critical gaps remain in the methodologies, tools, and integration frameworks currently in use**. Addressing these gaps is essential for developing robust, scalable, and user-friendly systems capable of supporting urban planners in real-world decision-making.

3.1.1 Fragmented Methodologies

One of the most persistent limitations across UHI research is the **fragmentation of methodologies**.

- **Remote sensing and GIS** approaches have been highly effective in detecting large-scale land surface temperature variations, but their outputs often remain at a resolution too coarse to capture fine-grained variations at the scale of individual buildings. They identify hotspots but cannot explain the **mechanisms driving localized heat accumulation**, such as the impact of specific building materials or orientations.
- **Computational Fluid Dynamics (CFD)** and microclimate models like ENVI-met offer highly detailed insights into airflow, shading, and thermal exchange. However, these models are **computationally expensive**, requiring significant expertise and resources. Their complexity often makes them unsuitable for quick, iterative planning decisions.
- **AI and machine learning models** demonstrate strong predictive power but often function as black-box systems. While they can detect correlations in large datasets, they provide little explanation of the **physical processes** underpinning the predictions. For planners and policymakers, this lack of interpretability limits their trust and applicability.

This fragmentation has created silos where each method excels in its own domain but fails to **holistically address UHI dynamics**.

3.1.2 Lack of Integration Between Tools

A second major gap lies in the **lack of integration** among existing modeling tools. Current workflows are often linear and disconnected:

- Remote sensing identifies temperature distributions.
- GIS overlays land-use data.
- CFD models simulate airflow.
- Energy modeling tools like EnergyPlus simulate building performance.

Yet, these systems are rarely connected into a **seamless pipeline**. Researchers or practitioners must manually extract, convert, and feed data from one system to another, introducing inefficiencies and data loss. For example, building geometries extracted from GIS often need manual processing before they can be used in CFD or thermal simulations.

This absence of integration results in duplicated efforts, inconsistent assumptions, and fragmented results. What is needed is an **end-to-end framework** that combines these methods within a unified simulation environment, reducing barriers for non-technical users.

3.1.3 Limited Metadata Utilization in Building Models

Another significant gap lies in the **underutilization of metadata** in building models. Most digital twins or 3D models represent geometry (shapes, textures, and layouts) but fail to include **physical and material properties** that are essential for thermal analysis. Without such properties, simulations cannot accurately capture how buildings absorb, retain, and release heat.

For UHI analysis, the following metadata should be embedded into building models:

- **Temperature (K)** – initial conditions or boundary values for simulation.
- **Area (m²)** – surface exposure for heat transfer.
- **Thickness (m)** – material depth affecting resistance.
- **Thermal Conductivity (W/m·K)** – capacity to conduct heat.
- **Specific Heat Capacity (J/kg·K)** – amount of energy required to raise material temperature.
- **Mass (kg)** – determining thermal inertia.

Current digital twin applications rarely enforce these requirements, leading to **oversimplified models**. Without metadata, simulations are forced to rely on default or generic assumptions, which reduces accuracy. This represents a clear opportunity: requiring metadata-rich building models as **a baseline input for UHI simulation pipelines**.

3.1.4 Static vs. Dynamic Analysis

Existing UHI models often operate under **static assumptions**.

- Remote sensing captures temperature at specific times but does not adapt to daily or seasonal fluctuations.
- CFD simulations are usually based on fixed boundary conditions and do not dynamically respond to weather changes.
- Many AI models are trained on historical datasets that may not generalize about future climate scenarios.

As a result, most analyses provide only **snapshots** of UHI conditions rather than capturing the **temporal dynamics** of urban thermal behavior. For cities experiencing rapid weather shifts or extreme climate variability, this is a major shortcoming.

A modern UHI analysis tool must incorporate **real-time environmental data** from APIs (e.g., temperature, humidity, wind speed, solar radiation) to provide adaptive and predictive outputs.

Without this capability, urban planners are left with outdated or static results that may not reflect actual conditions.

3.1.5 Decision-Making Support Gap

Even when existing tools generate accurate simulations, their outputs are often too **technical and inaccessible** for urban decision-makers. GIS maps, CFD flow diagrams, or AI predictive graphs require expert interpretation, which many city planners may not possess.
What is needed are **decision-oriented outputs**, such as:

- Roof A will reach 45 °C at midday under current conditions.
- Planting 10 trees in this block will reduce surface temperature by approximately 4 °C.
- Replacing asphalt with reflective paving can lower peak surface temperature by 6–7 °C.

This **translation of simulation results into actionable strategies** is rarely seen in current tools. Bridging this gap requires integrating data visualization, simplification, and recommendation features into the analysis framework.

Table 3-1 Comparison of current methods and tools in UHI measurement approaches vs. proposed simulation tool features.

Aspect	Current Methods / Tools	Limitations	Proposed Simulation Tool Features
Data Sources	GIS maps, manual field measurements, satellite imagery	Static, limited in accuracy; cannot integrate multiple datasets dynamically	Integrates **GIS data**, **Digital twin building models**, and **real-time weather APIs**
Visualization	2D maps, static heat	Lack interactivity; do not allow manipulation	**3D interactive viewer (React + Three.js)** with

	maps	of building geometry or metadata	model upload, drag, and rotation
Solar Exposure Modeling	Manual solar radiation charts specialized proprietary tools like ENVI-met	Requires advanced expertise; high cost	Uses **SunCalc** for dynamic solar position and exposure calculation
Weather Integration	Static weather datasets or government archives	Outdated, cannot simulate real-time scenarios	Fetches **real-time temperature, humidity, and wind speed** using weather APIs
Simulation Tools	Proprietary software (ANSYS Fluent, ENVI-met, EnergyPlus)	High cost, steep learning curve, inaccessible to local researchers	**MATLAB Simulink with Simscape Thermal**, automated pipeline from metadata
User Accessibility	Requires domain experts and powerful hardware	Not suitable for city planners, students, or quick evaluations	**Web-based interface** for interactive simulation and analysis
Scenario Testing	Manual and time-consuming	Limited ability to test alternative urban designs quickly	Allows **"what-if" simulations** (e.g., adding vegetation, changing materials)
Cost &	Expensive, academic	Not accessible for	Fully **open-source** +

Licensing	licenses limited	most developing countries	**academic-friendly MATLAB pipeline**
Output Insights	Static reports and charts	Lack actionable recommendations	Provides **quantitative results + AI-generated mitigation strategies**

The literature highlights substantial progress in individual areas of UHI research but shows a clear lack of **holistic, integrated solutions**. To summarize, the gaps include:

1. Fragmentation of methodologies (GIS, CFD, AI rarely connected).
2. Absence of integrated, user-friendly pipelines.
3. Metadata-poor building models that limit thermal accuracy.
4. Static rather than dynamic simulation approaches.
5. Lack of decision-making support for non-expert users.

These gaps collectively define the motivation for this research, which aims to design an **integrated digital twin–based simulation tool** to address these limitations.

3.2 Problem Statement

Urban areas are becoming increasingly vulnerable to the Urban Heat Island effect due to rapid urbanization, population growth, and climate change. The resulting rise in ambient temperature creates challenges such as:

- Increased energy demand for cooling,
- Public health risks due to heat stress, and
- Reduced overall urban livability.

Although numerous tools exist for studying UHI, their current limitations make them **insufficient for real-world urban planning**:

- GIS and remote sensing capture large-scale heat patterns but fail at building-level resolution.
- CFD and microclimate models provide accuracy but are computationally expensive and not widely accessible.
- AI models predict well but lack transparency and physical interpretability.
- Building models lack required metadata, reducing simulation reliability.
- Few platforms integrate real-time weather data, reducing adaptability.

Problem Statement (formal):
There is currently no integrated simulation tool that combines GIS datasets, metadata-rich 3D building models, real-time weather inputs, and physics-based thermal modeling to predict Urban Heat Island intensity at both building and neighborhood levels. The lack of such a framework prevents planners and policymakers from effectively testing and implementing UHI mitigation strategies in dynamic urban environments.

3.3 Hypothesis

This research is guided by the hypothesis that:

"An integrated digital twin–based simulation tool, combining GIS data, metadata-enriched 3D building models, weather API inputs, and MATLAB Simscape Thermal simulations, can accurately predict Urban Heat Island intensity at building and neighborhood scales, and translate results into actionable insights that support effective UHI mitigation planning."

Hypothesis Justification

1. **Integration Improves Accuracy:** By combining GIS, weather data, and building metadata, the model captures multiple variables simultaneously, reducing reliance on assumptions.
2. **Metadata-Rich Models Enhance Reliability:** Including parameters such as conductivity, specific heat, and mass ensures that simulations approximate real-world thermal responses.
3. **Dynamic Inputs Increase Practical Utility:** Real-time weather APIs enable the model to adapt, making it more relevant to actual urban conditions.
4. **Actionable Outputs Facilitate Adoption:** If outputs are expressed in practical terms (e.g., temperature reduction from greening strategies), adoption by city planners increases.

Validation of this hypothesis will involve developing the prototype simulation tool, running controlled experiments with varying material and environmental conditions, and comparing outputs to both **theoretical expectations** and **documented results from prior studies**.

4 OBJECTIVES

In the context of this research, the objectives are formulated to directly address the limitations identified in the literature review and the gaps outlined in the previous chapter. The Urban Heat Island (UHI) phenomenon is complex and multi-dimensional, requiring a multi-pronged approach that integrates geospatial analysis, building-level thermal simulation, and real-time environmental data. Therefore, the objectives of this research emphasize both **integration** and **practical utility**, ensuring that the outcomes are not only technically robust but also useful to planners, engineers, and policymakers.

4.1 Main Objective

The main objective of this research is:

"To develop an integrated simulation tool that combines GIS data, metadata-enriched 3D digital twins, real-time weather inputs, and MATLAB Simscape Thermal modeling to accurately simulate and visualize Urban Heat Island effects at both building and neighborhood scales, while providing decision-support outputs for sustainable urban planning."

This objective is directly motivated by the research gap identified, particularly the need for an end-to-end, metadata-driven, and adaptive framework for UHI analysis.

4.2 Sub-Objectives

To achieve the main objective, the following **sub-objectives** have been defined:

1. **To design a metadata-enriched 3D building model framework**

   This sub-objective focuses on establishing a standard for digital twins that go beyond geometry and textures by embedding critical thermal parameters. These include *temperature (K), surface area (m²), thickness (m), thermal conductivity (W/m·K), specific heat capacity (J/kg·K), and mass (kg)*. Ensuring that these values are incorporated at the model creation stage will enable accurate thermal simulations and overcome one of the key shortcomings of current digital twin applications.

2. **To integrate GIS and remote sensing data for spatial UHI analysis**

   Urban heat cannot be analyzed at the building scale alone; neighborhood-scale environmental variables must also be considered. This sub-objective involves the use of GIS datasets (roads, vegetation, water bodies, and building footprints) and remote sensing imagery (such as Sentinel 2 and Landsat 8) to contextualize building-level simulations within larger spatial patterns. By combining geospatial datasets with thermal simulations, the tool will provide a more holistic picture of UHI dynamics.

3. **To incorporate real-time weather data inputs into the simulation pipeline**

   Static simulations often fail to reflect the dynamic nature of UHI. Therefore, this sub-objective emphasizes linking the system with real-time weather APIs. Parameters such as temperature, humidity, wind speed, and solar radiation will be automatically fetched and fed into MATLAB simulations. This will enable adaptive modeling that reflects current and forecasted conditions, significantly enhancing the tool's predictive relevance.

4. **To simulate thermal behavior using MATLAB Simscape Thermal**

The core computational task of this research is the application of MATLAB Simscape Thermal for modeling heat transfer in building components. This includes conduction through walls, convection at surfaces, and radiation from sunlight exposure. By parameterizing models with metadata from the digital twin, the simulations will yield accurate predictions of surface and internal building temperatures under varying environmental scenarios.

5. To develop a visualization and decision-support interface

Complex outputs must be translated into formats understandable by non-technical users. This sub-objective entails creating a user interface, built on React and Three.js, where simulation results can be visualized in 3D, annotated with exposure values, and exported into simplified reports. The interface will allow users to compare alternative design or mitigation strategies, such as adding greenery, adjusting building orientation, or changing roof materials.

6. To validate the integrated tool against documented UHI patterns

Validation is crucial to establishing the credibility of the proposed framework. This sub-objective involves comparing simulation outputs with documented case studies and satellite-based temperature patterns from the literature. Discrepancies will be analyzed to refine metadata requirements, simulation parameters, and visualization strategies. This iterative validation will ensure that the tool not only demonstrates theoretical potential but also practical reliability.

Together, these objectives create a structured roadmap for research. The **main objective** provides the overarching vision, while the **sub-objectives** break it down into actionable steps, each tackling a specific aspect of the research gap. The metadata framework addresses the lack of detailed building-level models, the GIS and weather inputs tackle the issue of contextual integration, and the MATLAB simulations provide the physics-based backbone of the system.

5 METHODOLOGY

5.1 Introduction

The methodology of this research outlines the systematic approach taken to design and implement the proposed simulation tool for Urban Heat Island (UHI) analysis. This chapter details the **overall system design**, the **data acquisition process**, the **modeling and simulation pipeline**, and the **software development practices** followed. By combining GIS data, metadata-enriched 3D models, real-time weather inputs, and MATLAB Simscape Thermal simulations, the system is designed to deliver both high-fidelity analysis and practical decision-support outputs.
The methodology is divided into several stages, each addressing a component of the research objectives:

1. Import and process GIS data.
2. Metadata assignment for 3D building models.
3. Development of digital twin models in Blender.
4. Integration of real-time environmental data through weather APIs.
5. Simulation of sunlight exposure using SunCalc.
6. Thermal simulation pipeline in MATLAB Simulink.
7. System architecture and workflow diagrams.
8. Software Development Life Cycle (SDLC) approach.

5.2 Overall System Design

The system is designed as a modular pipeline where each component contributes to the overall UHI simulation process.

- **Input Layer:**
  - GIS datasets (roads, vegetation, water bodies, and building footprints).
  - 3D building models (GLB/STL format with embedded metadata).
  - Real-time environmental inputs (temperature, humidity, wind speed, and solar radiation).
- **Processing Layer:**
  - Metadata extraction and validation.
  - Integration of spatial data with building models.
  - Sunlight exposure calculation using SunCalc.
  - Thermal behavior simulation via MATLAB Simscape Thermal.
- **Output Layer:**
  - Spatial visualization in React + Three.js frontend.
  - Temperature distribution across building components.
  - Decision-support outputs for planners (e.g., roof overheating risk, effect of greening).

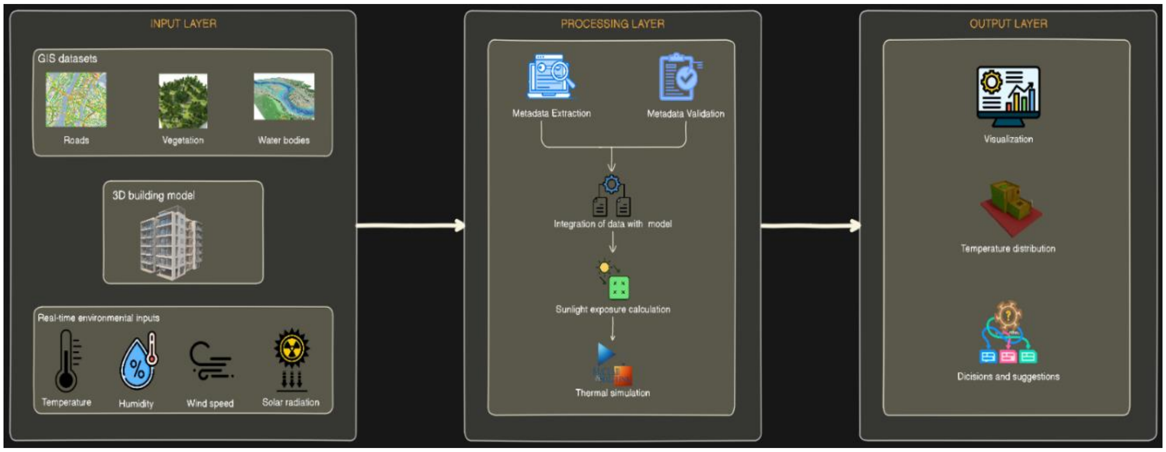


Figure 5-1 Overall System Architecture Diagram showing Input → Processing → Output flow.

5.3 GIS Data Import and Metadata Assignment

5.3.1 GIS Data Sources

For this research, **Google Map tiles** were used instead of OpenStreetMap (OSM) to extract spatial information such as building footprints, road networks, vegetation distribution, and water bodies. Google Maps provides **high-resolution, globally accessible map tiles** that allow for more detailed visualization compared to many open-source datasets. In addition, **Sentinel-2** and **Landsat-8** satellite imagery were used to analyze surface temperature and vegetation cover. Together, these datasets provide both **high spatial detail (Google Maps)** and **thermal-environmental context (Sentinel-2 and Landsat-8)**, ensuring a comprehensive urban heat island (UHI) analysis at both building and neighborhood scales.

Table 5-1 Visual comparison of Google Maps (high-resolution city blocks) vs. Sentinel-2 (urban land cover) vs. Landsat-8 (thermal hotspots)

Data Source	Resolution / Scale	Data Type	Usage in Research
Google Map Tiles	~0.5–2 m (varies by zoom level)	Vector + Raster (roads, buildings, land use, water bodies, vegetation overlays)	Extraction of building footprints, road layouts, vegetation coverage, and water body mapping.
Sentinel-2	10–20 m (visible & NIR bands)	Multispectral satellite imagery (13 bands)	Land Surface Temperature (LST) approximation, vegetation indices (NDVI), large-scale UHI mapping.
Landsat-8	30 m (multispectral), 100 m (thermal)	Multispectral + Thermal Infrared (TIRS)	Detailed thermal mapping, validation of UHI hotspots, calibration of surface temperature data.

5.3.2 Import Process

- Google Map tiles data is imported using **BlenderGIS plugin**, allowing roads, water bodies, and vegetation layers to be extracted.
- Satellite images are overlaid to provide temperature baselines for validation.

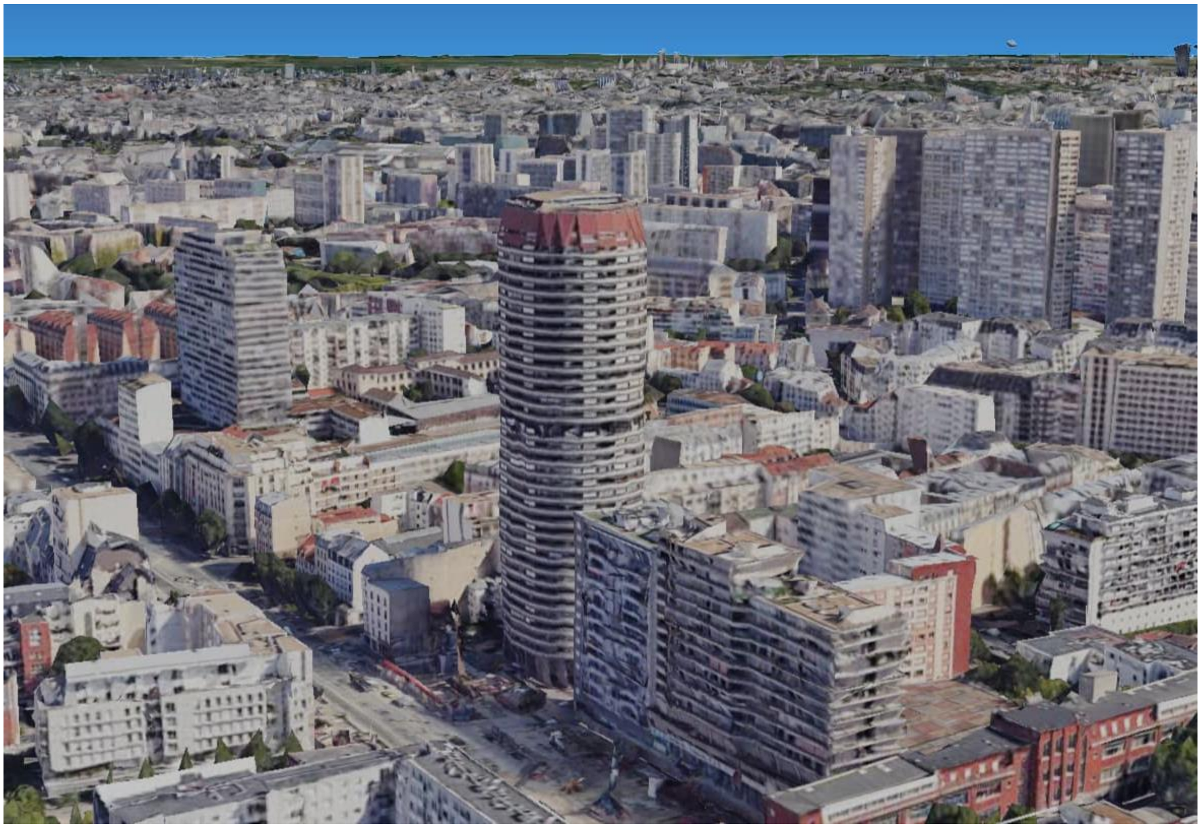


Figure 5-2 GIS data extracted using BlenderGIS plugin

5.3.3 Metadata Assignment

To enable accurate simulation of thermal behavior, it is not sufficient to model buildings based solely on their geometry. While geometric details such as dimensions and orientation are important, the **thermal response of a structure is fundamentally determined by its physical and material properties**. Therefore, in this research, each building object in the digital twin is enriched with a set of metadata parameters that represent its thermal characteristics.

The metadata is attached to building components such as walls, roofs, and floors using **Blender's custom property system**, ensuring that each object carries not only its geometric identity but also its physical profile. This approach transforms a purely visual 3D model into a **simulation-ready digital twin**.

The following parameters were selected for embedding, as they directly influence heat transfer processes:

- **Temperature (K):**
  Defines the initial thermal state of the material in Kelvin. This is crucial for setting up boundary conditions in simulations, ensuring that the model reflects realistic starting points (e.g., daytime vs. nighttime conditions).

- **Area (m^2):**
  Represents the exposed surface area of the component. Heat exchange with the surrounding environment, whether through conduction, convection, or radiation, is directly proportional to the exposed surface. Larger areas, such as expansive rooftops, absorb and release more heat compared to smaller components.

- **Thickness (m):**
  Specifies the depth of the material layer. Thicker walls and roofs act as stronger barriers to heat flow, offering higher thermal resistance. Conversely, thinner materials transfer heat more quickly, influencing how fast a building heats up or cools down.

- **Thermal Conductivity (W/m·K):**
  Indicates the material's ability to conduct heat. Materials such as concrete or steel have high conductivity, allowing heat to pass through rapidly, while insulating materials such as wood or fiberglass exhibit low conductivity. Accurate assignment of this parameter is essential to distinguish between heat-absorbing and heat-retaining materials.

- **Specific Heat Capacity (J/kg·K):**
  Defines the amount of energy required to raise the temperature of a unit mass of the material by one Kelvin. Materials with high specific heat capacity, such as water or concrete, can store large amounts of heat and release it gradually. This property explains why some urban surfaces, like asphalt, retain heat into the evening, intensifying nighttime UHI effects.

- **Mass (kg):**
  Represents the overall weight of the building component, which directly contributes to **thermal inertia**. Heavier components can resist temperature fluctuations for longer periods, moderating rapid swings in heat levels. This factor is particularly important when simulating diurnal (day–night) cycles in urban environments.

Table 5-2 Metadata parameters and units and role in simulation

Parameter	Unit	Role in Simulation
Temperature (K)	Kelvin	Initial condition
Area (m²)	Square meters	Heat transfer surface
Thickness (m)	Meters	Thermal resistance
Conductivity (W/m·K)	W/m·K	Heat conduction
Specific Heat (J/kg·K)	J/kg·K	Energy storage capacity
Mass (kg)	Kilograms	Thermal inertia

This metadata is stored as **custom object properties** within Blender and later exported to CSV for MATLAB.

Table 5-3 Sample metadata extracted from model

Object Name	Thickness (m)	Density (kg/mÂ³)	Thermal_Conductivity (W/mÂ·K)	Specific_Heat_Capacity (J/kgÂ·K)	Area	Mass	Material_type	Wind_Speed	Sun_Exposure	Temperature	Humidity
ac_exhaust009	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64
ac_exhaust010	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64
ac_exhaust011	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64
ac_exhaust012	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64
ac_exhaust013	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64
ac_exhaust014	0.2	2700	167	896	0.25	0.05	metal	4.1111	99.1	40	64
ac_exhaust015	0.2	2700	167	896	0.25	0.05	metal	4.1111	90.8	40	64
Brown_Cement_Wall-front	0.02	2400	1.8	900	276	5.52	concrete	4.1111	29.6	40	64
Brown_Cement_Wall-left	0.02	2400	1.8	900	196	3.92	concrete	4.1111	67.3	40	64
Brown_Cement_Wall-top001	0.02	2400	1.8	900	300	6	concrete	4.1111	58.3	40	64
concrete_wall_base002	0.25	2400	1.8	900	87.5	21.875	concrete	4.1111	31.3	40	64

5.4 3D Modeling in Blender (Digital Twin)

The creation of a digital twin is a critical step in this research, as it bridges the gap between real-world building data and simulation environments. Blender, an open-source 3D modeling and visualization software, was selected for this purpose due to its flexibility, active community support, and ability to integrate with GIS data through specialized plugins such as **BlenderGIS**.

The process begins with the **import of building footprints** from GIS datasets. Using tools like OpenStreetMap (OSM) and Sentinel-derived layers, the two-dimensional building footprints are extracted and then extruded into three-dimensional structures. This extrusion step transforms flat polygons into volumetric building blocks, representing the geometry of walls, roofs, and floors with accurate spatial dimensions. The generated geometry maintains alignment with geographic coordinates, ensuring that the buildings can be placed within the broader context of urban terrain.

Once the geometry is created, the next step involves enriching the model with **material properties and thermal metadata**. Unlike traditional 3D models that only represent physical appearance (e.g., shape, color, or texture), the digital twin in this study is designed to embed simulation-ready data. Blender's **custom property system** is used to assign key thermal parameters such as *surface area (m^2), thickness (m), thermal conductivity (W/m·K), specific heat capacity (J/kg·K), and mass (kg)*. These parameters enable the model to behave not just as a visual representation of a building but also as an analytical unit capable of supporting physics-based thermal simulations. For instance, a wall component with higher thermal conductivity will demonstrate faster heat transfer during simulation, while one with greater specific heat capacity will exhibit higher energy storage potential.

After embedding metadata, the models are prepared for integration with the simulation pipeline. To ensure compatibility with downstream tools, particularly MATLAB Simscape Thermal, the enriched models are **exported in GLB or STL formats**. These formats preserve geometry while maintaining links to the associated metadata, which is simultaneously exported as a structured CSV file. This dual-export mechanism ensures that while MATLAB processes numerical simulation data, the visual frontend (React + Three.js) can display the digital twin for **interactive visualization and result interpretation**.

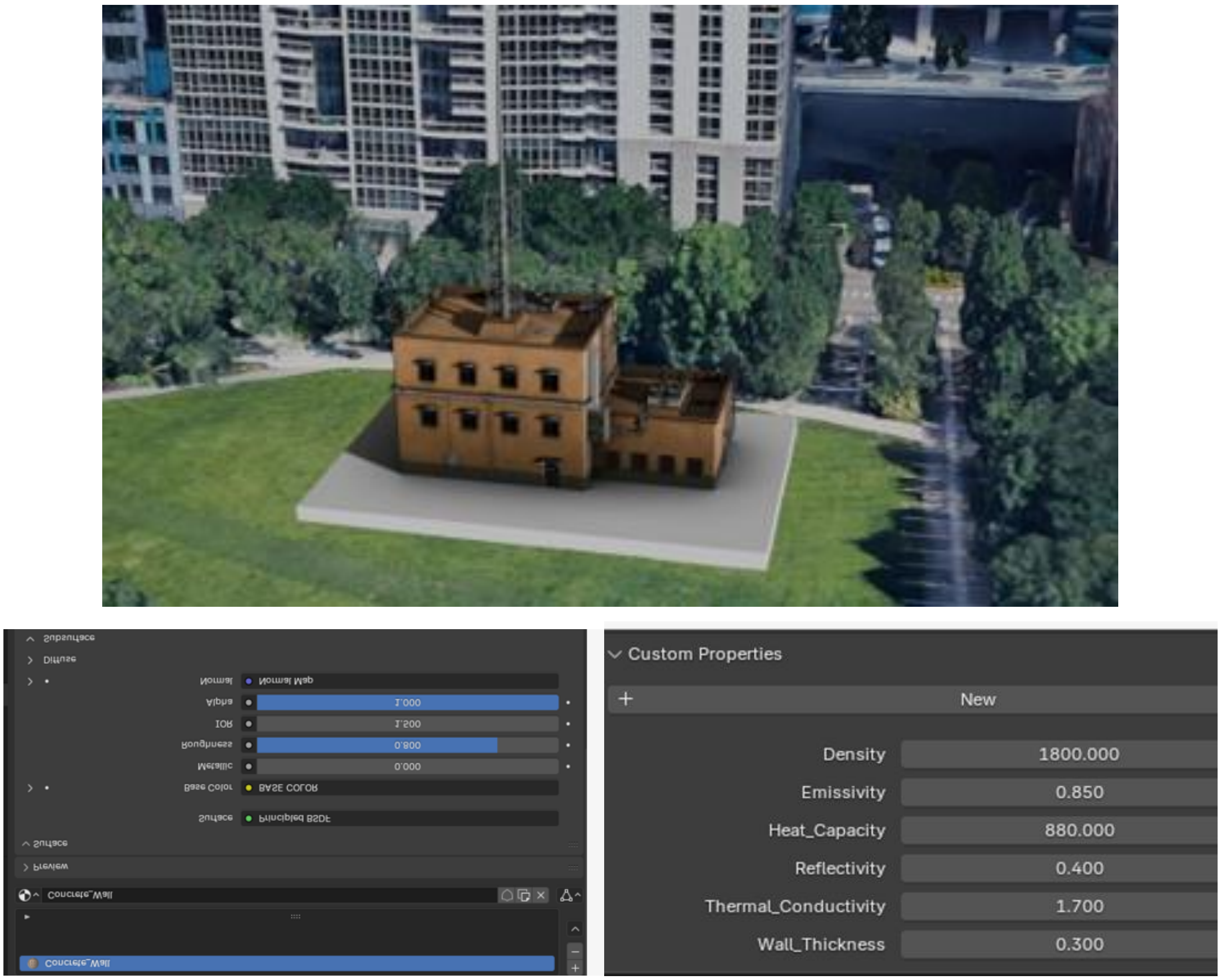


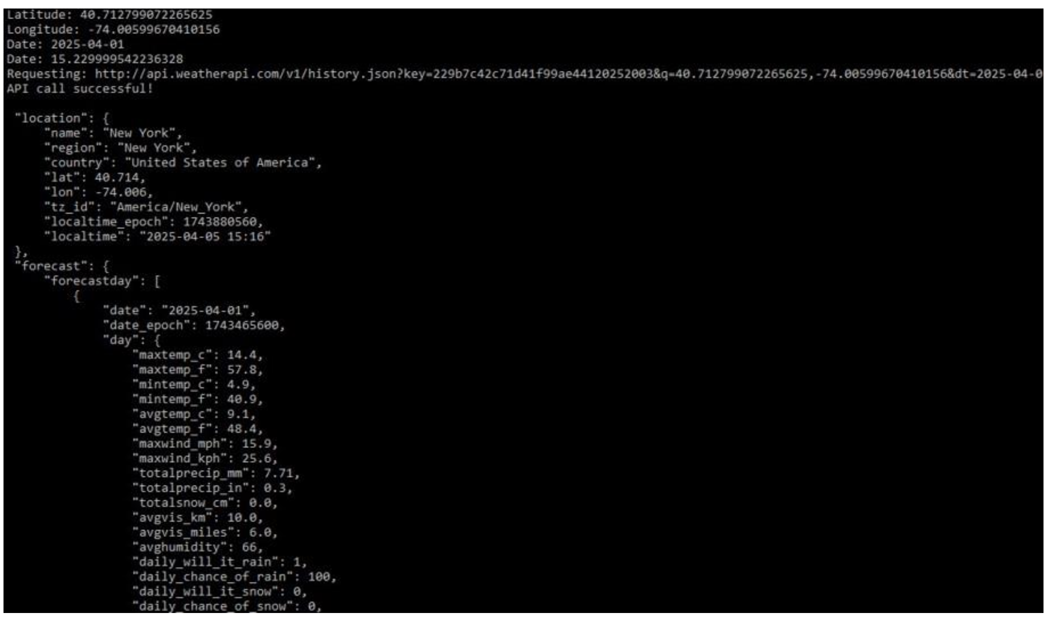
Figure 5-3 Example material tags and custom properties attached to wall of the building model.

This approach ensures that the digital twin not only represents spatial geometry but also **encodes thermal characteristics**, making it suitable for MATLAB simulations.

5.5 Weather API Integration

The **WeatherAPI** is used to fetch real-time and forecast weather data.

- Parameters imported:
  - Ambient Temperature (°C)
  - Humidity (%)
  - Wind Speed (m/s)
  - Solar Radiation (W/m², approximated)
- Integration method:
  - API requests are scheduled based on the simulation time frame and longitude latitude given by user.
  - Data is injected in to the csv file (metadata extracted from the model) with the new columns for every field (Humidity, Wind speed, Solar radiation, Temperature).



```
Latitude: 40.712799072265625
Longitude: -74.00599670410156
Date: 2025-04-01
Date: 15.229999542236328
Requesting: http://api.weatherapi.com/v1/history.json?key=229b7c42c71d41f99ae44120252003&q=40.712799072265625,-74.00599670410156&dt=2025-04-0
API call successful!

 "location": {
     "name": "New York",
     "region": "New York",
     "country": "United States of America",
     "lat": 40.714,
     "lon": -74.006,
     "tz_id": "America/New_York",
     "localtime_epoch": 1743880560,
     "localtime": "2025-04-05 15:16"
 },
 "forecast": {
     "forecastday": [
         {
             "date": "2025-04-01",
             "date_epoch": 1743465600,
             "day": {
                 "maxtemp_c": 14.4,
                 "maxtemp_f": 57.8,
                 "mintemp_c": 4.9,
                 "mintemp_f": 40.9,
                 "avgtemp_c": 9.1,
                 "avgtemp_f": 48.4,
                 "maxwind_mph": 15.9,
                 "maxwind_kph": 25.6,
                 "totalprecip_mm": 7.71,
                 "totalprecip_in": 0.3,
                 "totalsnow_cm": 0.0,
                 "avgvis_km": 10.0,
                 "avgvis_miles": 6.0,
                 "avghumidity": 66,
                 "daily_will_it_rain": 1,
                 "daily_chance_of_rain": 100,
                 "daily_will_it_snow": 0,
                 "daily_chance_of_snow": 0,
```


Figure 5-4 Sample weather API response

This allows the simulation pipeline to reflect **dynamic urban conditions**, rather than static snapshots.

5.6 Sunlight Exposure Simulation (SunCalc)

SunCalc is integrated to calculate **sun position and shadow casting** at specific times and locations.

- **Input:** Latitude, longitude, date, and time.
- **Output:** Azimuth, altitude, and shadow length.
- **Usage:** Determines the percentage of surface area exposed to direct sunlight.

The calculated sunlight exposure values are embedded into the building metadata, influencing heat gain calculations in MATLAB Simscape.

Equation 5-1 Sunlight Exposure Percentage (SEP)

$$SEP = \frac{A_{exposed}}{A_{total}} \times 100$$

Where $A_{exposed}$ is the sunlit surface area, and A_{total} is the total surface area of the component.

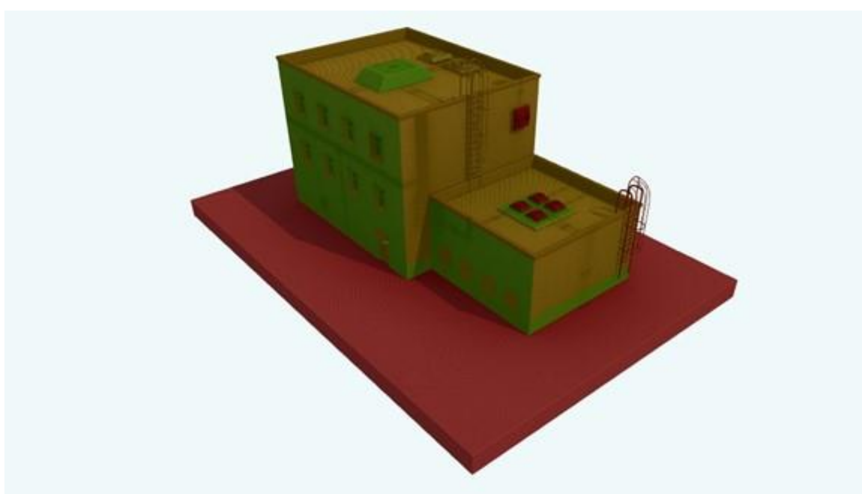


Figure 5-5 - Example Heatmap Overlay of building model after sunlight exposure calculation

5.7 MATLAB Simulink Simulation Pipeline

MATLAB Simscape Thermal is the core simulation engine.

- **Metadata Input:** CSV file containing building component properties.
- **Thermal Modeling:**
    - Conduction modeled using thermal resistance equations.
    - Convection modeled via surface heat transfer coefficients.
    - Radiation is modeled based on sunlight exposure.
- **Outputs:**
    - Temperature distribution across roof, walls, and floors.
    - Energy consumption estimates.
    - Heat storage capacity of different materials.

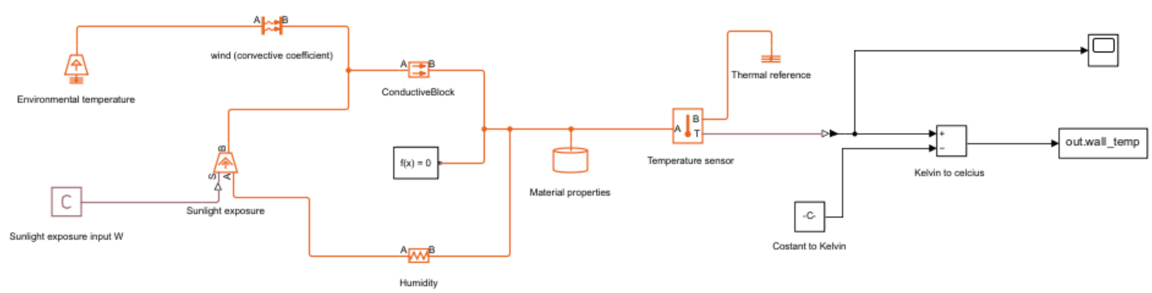


Figure 5-6 MATLAB Simulink block diagram showing data flow from CSV → Simscape Thermal blocks → output plots.

This pipeline enables both **static simulations (fixed conditions)** and **dynamic simulations (real-time weather integration)**.

Table 5-4 Sample output data after simulation

ObjectName	Thickness_m_	Density_kg_m	Thermal_Conductivity_W_m_K	Specific_Heat_Capacity_J_kg_K	Area	Mass	Material_	Wind_Speed	Sun_Exposure	Temperature	Humidity	Final_Temperature
ac_exhaust009	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64	38.80239521
ac_exhaust010	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64	38.80239521
ac_exhaust011	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64	38.80239521
ac_exhaust012	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64	38.80239521
ac_exhaust013	0.2	2700	167	896	0.25	0.05	metal	4.1111	100	40	64	38.80239521
ac_exhaust014	0.2	2700	167	896	0.25	0.05	metal	4.1111	99.1	40	64	38.81317365
ac_exhaust015	0.2	2700	167	896	0.25	0.05	metal	4.1111	98.8	40	64	38.81676647
Brown_Cement_Wall-front	0.02	2400	1.8	900	276	5.52	concrete	4.1111	29.6	40	64	36.71111111
Brown_Cement_Wall-left	0.02	2400	1.8	900	196	3.92	concrete	4.1111	67.3	40	64	32.52222222
Brown_Cement_Wall-top001	0.02	2400	1.8	900	300	6	concrete	4.1111	58.3	40	64	33.52222222

5.8 System Architecture & Flow Diagrams

The complete system architecture integrates all modules:

1. **Frontend (React + Three.js):** Visualization and user input.
2. **Backend (Node.js):** Manages API requests and file handling.
3. **Blender:** Preprocessing and metadata embedding.
4. **MATLAB:** Core thermal simulation.

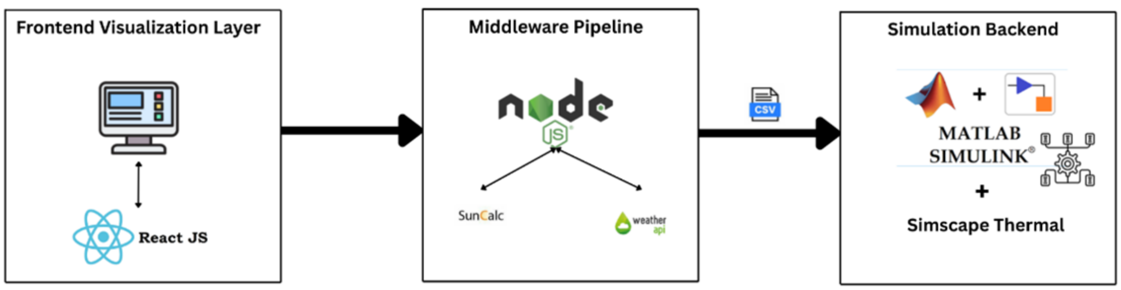


Figure 5-7 Three-tier architecture of the Urban Heat Island (UHI) Simulation Tool, comprising frontend visualization layer, middle pipeline, simulation backend technologies

5.9 SDLC Approach

This project follows the **Agile Software Development Life Cycle (SDLC)** approach, which emphasizes iterative development, modular design, and continuous feedback integration. Agile was selected over traditional linear approaches such as the Waterfall model because the system under development involves multiple interdependent modules (GIS data handling, metadata embedding, weather API integration, sunlight exposure calculations, and MATLAB simulation). Each of these modules requires flexibility, testing, and refinement as the system evolves. The Agile SDLC ensures that the project can adapt to changing requirements, incorporate stakeholder feedback, and progressively build towards the final simulation tool.

The following subsections detail each SDLC phase and its role in this research.

5.9.1 Requirements Gathering and Analysis

The requirement analysis phase focused on identifying the gaps in current Urban Heat Island

(UHI) modeling tools and defining the scope of the proposed solution. A thorough literature review revealed that existing tools suffer from limitations such as:

- Fragmented methodologies (GIS, CFD, and AI tools rarely integrated).
- Metadata-poor building models lacking essential thermal parameters.
- Static simulations that fail to reflect real-time weather dynamics.
- Lack of actionable outputs for urban planners and decision-makers.

Based on these findings, the project requirements were defined as:

1. The system must integrate **GIS data** for environmental context.
2. Building models must include **metadata properties** (temperature, area, thickness, conductivity, specific heat, and mass).
3. The tool must use a **weather API** to incorporate real-time inputs.
4. Sunlight exposure simulation must be integrated to capture diurnal variations.
5. MATLAB Simscape Thermal must be used as the **core simulation engine**.
6. Results must be visualized in an accessible **frontend interface**.

This requirement sets the foundation for the design and implementation strategy.

5.9.2 Design

The design phase translated the requirements into a **modular system architecture**. The architecture was divided into three primary layers:

- **Input Layer**: Handles GIS data import, building models, and weather API inputs.
- **Processing Layer**: Conducts metadata validation, sunlight exposure simulation, and thermal behavior modeling in MATLAB.
- **Output Layer**: Provides visualization through React + Three.js and decision-support summaries.

The modular design ensures that each component (e.g., GIS import, MATLAB simulation) can be developed, tested, and refined independently. Data flow diagrams and system architecture diagrams were created to illustrate the interaction between components, ensuring smooth integration. This design approach reduces complexity, promotes scalability, and allows for future extensions, such as integrating AI-based predictive models.

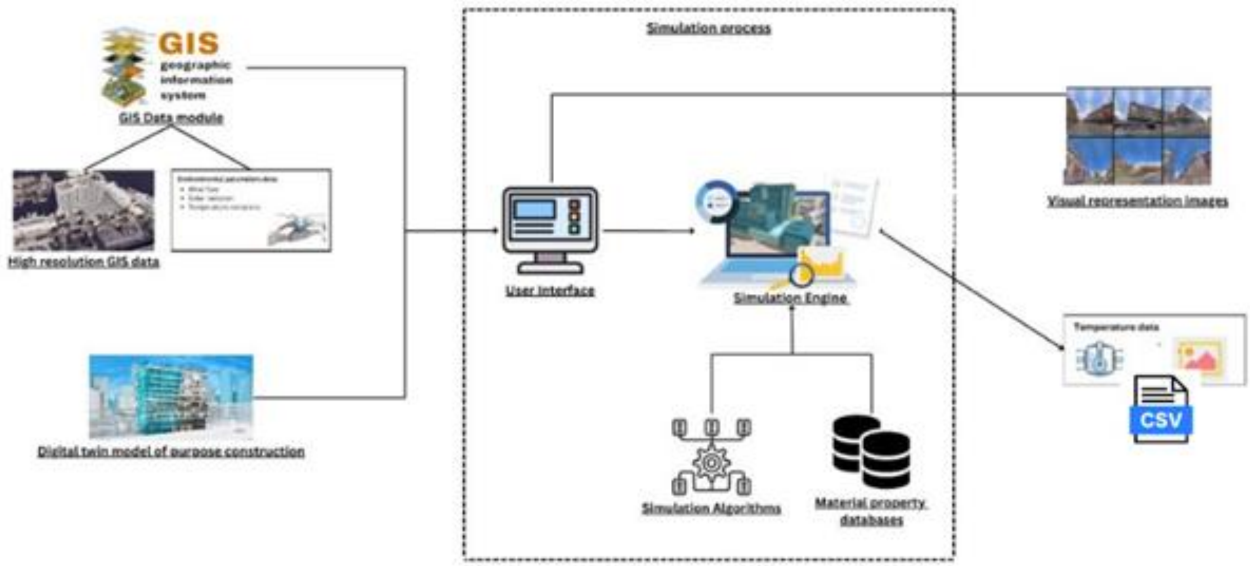


Figure 5-8 High level component diagram

5.9.3 Implementation

The implementation phase focused on **incremental development of system modules**. Following Agile practices, each module was treated as a separate sprint:

1. **Sprint 1:** GIS data import and preprocessing using BlenderGIS and Google Map tiles data datasets.
2. **Sprint 2:** Metadata embedding for building models, ensuring thermal parameters were attached as custom properties.
3. **Sprint 3:** Sunlight exposure simulation using SunCalc integrated into the pipeline.
4. **Sprint 4:** Weather API integration for real-time environmental data retrieval.
5. **Sprint 5:** MATLAB Simscape Thermal modeling pipeline, accepting metadata CSVs as inputs.
6. **Sprint 6:** Frontend visualization interface with React + Three.js.

This incremental implementation ensured that each feature could be tested and validated before integration with the rest of the system. By adopting Agile sprints, development risks were minimized, and early results could be demonstrated to stakeholders for feedback.

5.9.4 Testing

Testing was carried out at two levels:

- **Module-Level Testing (Unit Testing):** Each module (e.g., weather API handler, SunCalc integration, MATLAB pipeline) was tested independently. For instance, the SunCalc module was validated by comparing its shadow calculations with Blender’s native sun simulation.
- **Integration Testing:** Once modules were individually validated, they were connected into a full pipeline. Integration testing ensured that the metadata exported from Blender matched the input requirements of MATLAB, and that weather API data flowed seamlessly into simulations.

Additionally, outputs were cross-checked against documented UHI case studies from the literature to ensure validity. This testing phase guaranteed the **accuracy, reliability, and interoperability** of the system.

Accurate UHI simulation requires multi-source data integration.

5.9.5 Deployment

The deployment phase focused on making the system accessible to end-users in a cloud environment. The architecture was containerized using **Docker**, ensuring portability and scalability. The MATLAB simulation engine was deployed in conjunction with the backend server to allow remote execution of thermal models.

The frontend (React + Three.js) was hosted on a cloud-based platform, enabling users to upload 3D building models, view sunlight exposure, fetch weather data, and trigger simulations. Deployment also involved configuring security measures such as IAM roles, environment variables for API keys, and HTTPS encryption to ensure data privacy.

This deployment ensures that the system can be accessed across devices, supporting planners, engineers, and researchers regardless of location.

5.9.6 Maintenance

The final phase of the SDLC focuses on **sustaining and enhancing the system** beyond its initial deployment. Given the evolving nature of climate data, urban growth, and technological advancements, continuous updates are essential. Maintenance includes:

- **Bug Fixes:** Identifying and resolving issues that may arise in API connections, metadata handling, or simulation accuracy.
- **System Updates:** Upgrading to newer versions of MATLAB, Blender, and React as they are released.
- **Feature Enhancements:** Expanding the system with AI/ML modules for predictive analysis or integrating additional environmental factors such as humidity-driven heat stress models.
- **User Feedback:** Iteratively refining the interface and outputs based on stakeholder input to improve usability.

By planning for long-term maintenance, the system is positioned not as a one-off research prototype but as a sustainable, extensible tool for UHI analysis.

The process integrates GIS datasets, metadata-rich 3D modeling, real-time weather data, sunlight exposure calculations, and MATLAB Simscape Thermal simulations into a unified pipeline. The system architecture ensures modularity and scalability, while the Agile SDLC approach ensures iterative refinement and validation.

6 Project Requirements

Defining project requirements is a critical step in ensuring that the proposed simulation tool meets both the technical specifications and the expectations of its end-users. This chapter outlines the **functional**, **non-functional**, **system**, and **user requirements** that guide the design and implementation of the Urban Heat Island (UHI) simulation tool. These requirements ensure that the system is technically feasible, user-friendly, and capable of supporting sustainable urban planning initiatives.

6.1 Functional Requirements

Functional requirements describe the specific features and behaviors that the system must perform. They are derived from the research objectives and the identified gaps in existing UHI modeling tools.

1. **GIS Data Import**
    - The system allows importing GIS datasets such as roads, vegetation, water bodies, and building footprints.
    - The system should support multiple formats (e.g., GeoJSON, Shapefile, OSM, Google Map Tiles).
2. **3D Building Model Handling**
    - The system shall enable uploading of 3D building models in GLB or STL format.
    - The system shall assign and validate metadata (temperature, area, thickness, conductivity, heat capacity, and mass) for each building component.
3. **Weather API Integration**
    - The system shall fetch real-time weather parameters (temperature, humidity, wind speed, solar radiation) using an external API.
    - The system should process and map API responses into MATLAB-compatible variables.

4. **Sunlight Exposure Simulation**
    - The system shall compute sunlight position and exposure using SunCalc based on location and time.
    - The system shall calculate the percentage of building surfaces exposed to direct sunlight.
5. **Thermal Simulation (MATLAB Simscape)**
    - The system shall simulate conduction, convection, and radiation processes for each building component.
    - The system should output temperature distributions for roofs, walls, and floors.
6. **Visualization and Reporting**
    - The system shall provide a 3D visualization interface (React + Three.js) for simulation results.
    - The system shall generate reports with summarized outputs (e.g., surface temperature differences, mitigation impact).

6.2 Non-Functional Requirements

Non-functional requirements define system qualities such as performance, scalability, security, and usability.

1. **Performance**
    - The system shall process and display simulation results within acceptable time frames (≤ 5 minutes for standard building datasets).
2. **Scalability**
    - The system shall handle simulations at both single-building and neighborhood scales.
    - The architecture shall support cloud deployment for larger datasets.
3. **Security**
    - The system shall ensure secure API communication using HTTPS.
    - Sensitive data (user uploads, results) shall be stored with access control policies.
4. **Usability**
    - The user interface shall be intuitive and provide clear guidance for non-technical users.
    - Outputs shall be presented in both graphical (3D visualization) and numerical (tables, CSV) forms.
5. **Reliability**
    - The system shall recover gracefully from failed API calls or incomplete uploads.
    - Backup and retry mechanisms shall be implemented for external data dependencies.

6.3 System Requirements

System requirements specify the hardware and software platforms necessary to run the proposed tool.

6.3.1 Hardware Requirements

- **Minimum Configuration (Local Testing):**
  - Processor: Intel i5 / AMD Ryzen 5 or equivalent
  - RAM: 8 GB
  - Storage: 250 GB SSD
  - GPU: Integrated graphics sufficient for basic 3D rendering
- **Recommended Configuration (Full Simulation):**
  - Processor: Intel i7 / AMD Ryzen 7 or higher
  - RAM: 16 GB+
  - Storage: 500 GB SSD
  - GPU: NVIDIA GTX/RTX series or AMD equivalent

6.3.2 Software Requirements

- **Frontend:** React, Three.js, Bootstrap
- **Backend:** Node.js (v14+), Express.js
- **3D Modeling:** Blender 3.x with BlenderGIS plugin
- **Simulation Engine:** MATLAB R2022b+ with Simscape Thermal toolbox
- **Database (optional for persistence):** Firebase
- **Cloud Services:** AWS ECS/Fargate, S3 for storage (future deployment)

6.4 User Requirements

User requirements capture what the **end-users** expect from the system. These are primarily urban planners, architects, environmental researchers, and policymakers.

1. **Ease of Use**
    - Users shall be able to upload building models without needing technical expertise in Blender or GIS.
    - The system shall automatically validate and prompt users to add missing metadata.
2. **Accessible Results**
    - Users shall view simulation outputs in an interactive 3D interface.
    - Results shall be presented with clear labels (e.g., “Roof Surface Temperature: 45 °C”).
3. **Scenario Testing**
    - Users shall test “what-if” scenarios (e.g., adding vegetation, changing roof material) to compare UHI mitigation strategies.
    - Users shall export results for policy reports in PDF or CSV format.
4. **Collaboration and Decision Support**
    - The system shall allow multiple users (e.g., planner + researcher) to access the same dataset.
    - The system shall present outputs in a decision-support style (e.g., “Replacing asphalt with reflective pavement may reduce temperatures by ~6 °C”).
5. **Reliability of Information**
    - Users require accurate and validated simulations, aligning results with known satellite/field data.
    - The system shall clearly indicate the assumptions and limitations of each simulation run.

7 FRONTEND / SYSTEM DESIGN

7.1 Frontend Design (React + Three.js)

The front end serves as the user-facing interface that enables users to interact with the simulation tool. It was implemented using React for the UI framework and Three.js for 3D model rendering.

7.1.1 Key Features

- **File Upload Module**
  - Users can upload building models in .glb or .stl format.
  - Metadata validation is performed after upload.
- **3D Viewer with Three.js**
  - Displays building and GIS models.
  - Enables navigation (pan, zoom, rotate) and object selection.
  - Includes **TransformControls** to reposition GIS objects relative to building models.
- **Sunlight Exposure Overlay**
  - Integrates with SunCalc results to show real-time shadow effects.
  - Highlights building areas exposed to sunlight vs shaded areas.
- **Dashboard & Controls**
  - User-friendly dashboard showing simulation options.
  - Control panel for selecting weather data time frame, running MATLAB simulations, and exporting results.

7.1.2 Implementation Details

- Developed with **React functional components**.
- Used **Tailwind CSS** for styling and **Framer Motion** for animations.
- 3D rendering handled by **react-three-fiber (Three.js wrapper)**.
- File uploads managed using **React Dropzone** and stored temporarily in backend storage.

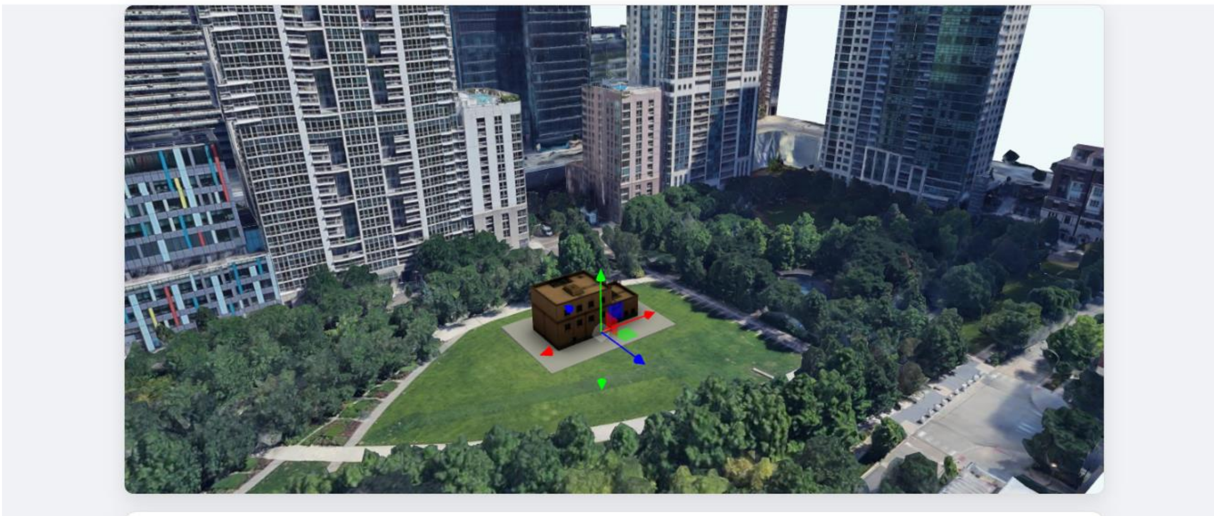


Figure 7-1 Web application frontend 1

Figure 7-2 Web Application frontend 2

Figure 7-3 Simulation output 1

Figure 7-4 Simulation output 2

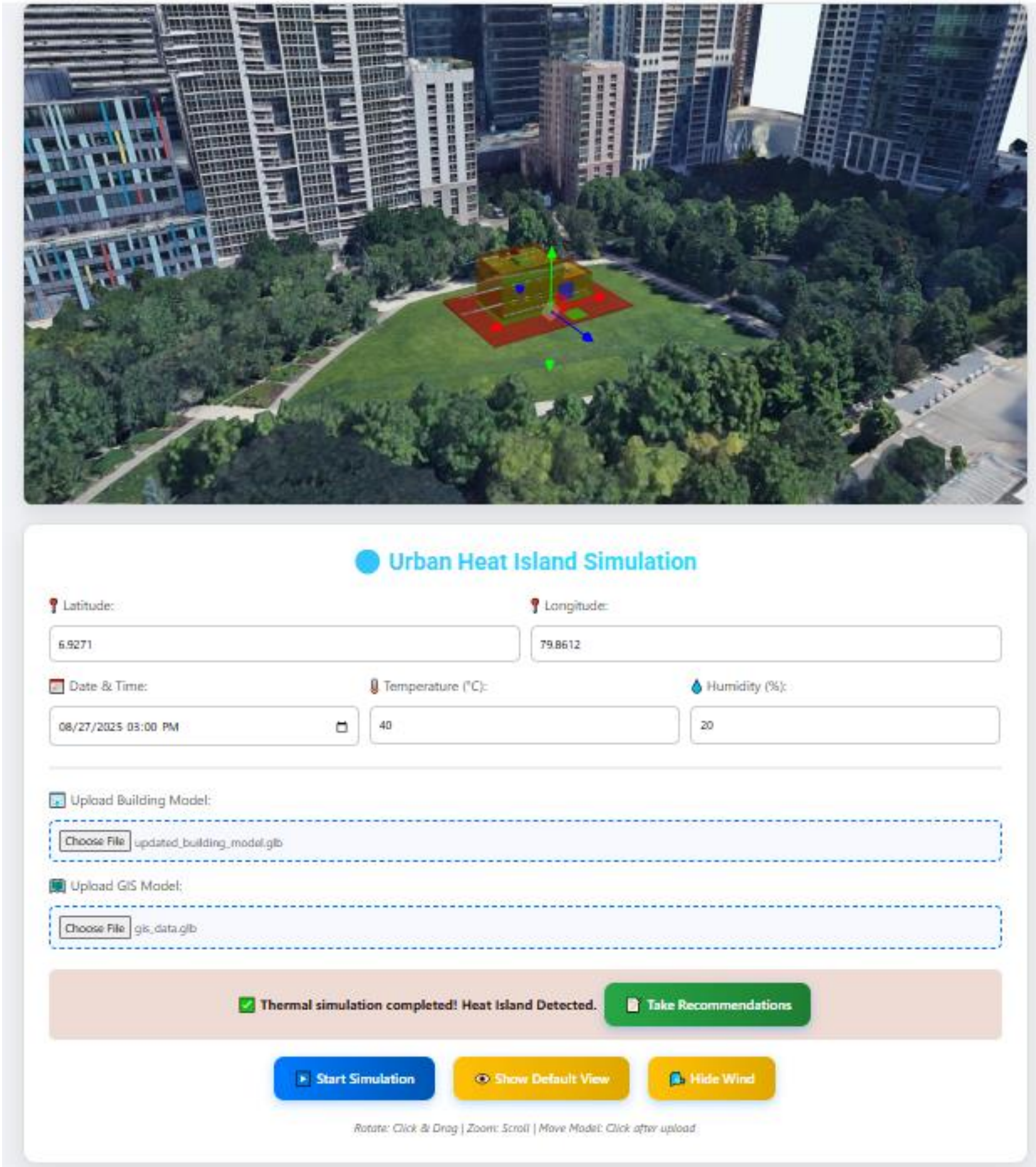


Figure 7-5 Full web application front end

8 EXPERIMENTS AND RESULTS

This chapter presents the experiments conducted to evaluate the performance, accuracy, and usability of the proposed UHI simulation tool. The evaluation was designed to assess:

1. The ability of the **MATLAB Simscape Thermal pipeline** to simulate heat transfer under varying material and weather conditions.
2. The effectiveness of the **SunCalc sunlight exposure integration** in estimating solar gains.
3. The responsiveness and usability of the **frontend–backend pipeline** in handling real-world building models with metadata.
4. The alignment of simulation outputs with documented UHI patterns and existing models.

Experiments were organized into three categories: **model-level testing** (material & weather variation), **system-level testing** (pipeline performance), and **validation testing** (comparison with known UHI benchmarks).

8.1 Experimental Setup

The system was tested in both **controlled and semi-real-world environments**.

- **Hardware:** Intel i7, 16 GB RAM, NVIDIA GTX 1650 GPU.
- **Frontend:** React + Three.js interactive viewer.
- **Backend:** Node.js server with CSV pipeline.
- **Simulation Engine:** MATLAB Simscape Thermal running conduction, convection, and radiation models.
- **Datasets:** GIS + building models (Blender-generated with metadata), real-time weather API (temperature, humidity, wind, solar radiation).

8.1.1 Simulation Workflow

1. Upload 3D building model with metadata.
2. Extract Area, Thickness, Conductivity, Heat Capacity, Mass.
3. Fetch weather data + compute SunCalc exposure.
4. Generate CSV → MATLAB input.
5. Run thermal simulations in MATLAB Simscape.
6. Visualize outputs in React dashboard.

8.2 Simulation Scenarios

Experiments were carried out across **different materials** and **weather conditions** to evaluate system adaptability.

8.2.1 Material-Based Scenarios

Three material profiles were simulated:

- **Concrete wall:** High thermal mass, conductivity 1.2 W/m·K.
- **Brick wall:** Moderate thermal mass, conductivity 0.72 W/m·K.
- **Wood panel:** Low thermal mass, conductivity 0.12 W/m·K.

Table 8-1 Material properties dataset used for experiment

	A	B	C	D	E	F	G	H
1	Object Name	Thickness (m)	Density (kg/mÂ³)	Thermal_Conductivity (W/mÂ·K)	Specific_Heat_Capacity (J/kgÂ·K)	Area	Mass	Material_type
2	ac_exhaust009	0.2	2700	167	896	0.25	0.05	metal
3	ac_exhaust010	0.2	2700	167	896	0.25	0.05	metal
4	ac_exhaust011	0.2	2700	167	896	0.25	0.05	metal
5	ac_exhaust012	0.2	2700	167	896	0.25	0.05	metal
6	ac_exhaust013	0.2	2700	167	896	0.25	0.05	metal
7	ac_exhaust014	0.2	2700	167	896	0.25	0.05	metal
8	ac_exhaust015	0.2	2700	167	896	0.25	0.05	metal
9	Brown_Cement_Wall-front_	0.02	2400	1.8	900	276	5.52	concrete
10	Brown_Cement_Wall-left_ri	0.02	2400	1.8	900	196	3.92	concrete
11	Brown_Cement_Wall-top00	0.02	2400	1.8	900	300	6	concrete
12	concrete_wall_base002	0.25	2400	1.8	900	87.5	21.875	concrete
13	concrete_wall_base003	0.25	2400	1.8	900	16.9	4.225	concrete
14	window_frame008	0.025	2700	278	900	0.115	0.002875	steel
15	window_glass008	0.004	2230	1.12	830	0.26	0.00104	glass
16	window_frame009	0.025	2700	218	900	0.115	0.002875	steel
17	window_glass009	0.004	2230	1.12	830	0.26	0.00104	glass
18	window_frame010	0.025	2700	218	900	0.115	0.002875	steel

8.2.2 Weather-Based Scenarios

Simulations considered **diurnal variation** (day vs night) and **seasonal variation**.

- Hot summer day: 35 °C, clear sky, wind 2 m/s.
- Mild winter day: 18 °C, partly cloudy, wind 3 m/s.
- Humid monsoon day: 28 °C, humidity 85%, overcast

Table 8-2 Weather profiles used in simulation runs.

Object Name	Wind_Speed	Sun_Exposure	Temperature	Humidity
ac_exhaust009	6.3056	100	40	20
ac_exhaust010	6.3056	100	40	20
ac_exhaust011	6.3056	100	40	20
ac_exhaust012	6.3056	100	40	20
ac_exhaust013	6.3056	100	40	20
ac_exhaust014	6.3056	99.1	40	20
ac_exhaust015	6.3056	99.7	40	20
Brown_Cement_Wall-front_	6.3056	27.4	40	20
Brown_Cement_Wall-left_ri	6.3056	86.5	40	20
Brown_Cement_Wall-top00	6.3056	66.7	40	20
concrete_wall_base002	6.3056	31.3	40	20
concrete_wall_base003	6.3056	50	40	20
window_frame008	6.3056	59.8	40	20
window_glass008	6.3056	0	40	20
window_frame009	6.3056	59.8	40	20
window_glass009	6.3056	0	40	20

8.3 Simulation Results

Outputs show **component-wise temperature variation** (roof, wall, floor), SunCalc integration provided exposure percentages for each surface, and temperature **gradients** across building surfaces.

Table 8-3 Simulation output

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ObjectName	Thickness_m_	Density_kg_m__	Thermal_Conductivity_W_m_K_	Specific_Heat_Capacity_J_kg_K_	Area	Mass	Material_type	Wind_Speed	Sun_Exposure	Temperature	Humidity	FinalWallTemperature
2	ac_exhaust009	0.2	2700	167	896	0.25	0.05	metal	6.3056	100	40	20	38.80239521
3	ac_exhaust010	0.2	2700	167	896	0.25	0.05	metal	6.3056	100	40	20	38.80239521
4	ac_exhaust011	0.2	2700	167	896	0.25	0.05	metal	6.3056	100	40	20	38.80239521
5	ac_exhaust012	0.2	2700	167	896	0.25	0.05	metal	6.3056	100	40	20	38.80239521
6	ac_exhaust013	0.2	2700	167	896	0.25	0.05	metal	6.3056	100	40	20	38.80239521
7	ac_exhaust014	0.2	2700	167	896	0.25	0.05	metal	6.3056	99.1	40	20	38.81317365
8	ac_exhaust015	0.2	2700	167	896	0.25	0.05	metal	6.3056	99.7	40	20	38.80598802
9	Brown_Cement_Wall-front_bacl	0.02	2400	1.8	900	276	5.52	concrete	6.3056	27.4	40	20	36.95555556
10	Brown_Cement_Wall-left_right0	0.02	2400	1.8	900	196	3.92	concrete	6.3056	86.5	40	20	30.38888889
11	Brown_Cement_Wall-top001	0.02	2400	1.8	900	300	6	concrete	6.3056	66.7	40	20	32.58888889
12	concrete_wall_base002	0.25	2400	1.8	900	87.5	21.875	concrete	6.3056	31.3	40	20	-3.472222222
13	concrete_wall_base003	0.25	2400	1.8	900	16.9	4.225	concrete	6.3056	50	40	20	-29.44444444
14	window_frame008	0.025	2700	278	900	0.115	0.002875	steel	6.3056	59.8	40	20	39.94622302
15	window_glass008	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
16	window_frame009	0.025	2700	218	900	0.115	0.002875	steel	6.3056	59.8	40	20	39.93142202
17	window_glass009	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
18	window_frame010	0.025	2700	218	900	0.115	0.002875	steel	6.3056	51.6	40	20	39.94082569
19	window_glass010	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	25	40	20	39.10714286
20	window_frame011	0.025	2700	218	900	0.115	0.002875	steel	6.3056	59.8	40	20	39.93142202
21	window_glass011	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
22	window_frame012	0.025	2700	218	900	0.115	0.002875	steel	6.3056	45.1	40	20	39.94827982
23	window_glass012	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
24	window_frame013	0.025	2700	218	900	0.115	0.002875	steel	6.3056	45.1	40	20	39.94827982
25	window_glass013	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
26	window_frame014	0.025	2700	218	900	0.115	0.002875	steel	6.3056	45.1	40	20	39.94827982
27	window_glass014	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
28	window_frame015	0.025	2700	218	900	0.115	0.002875	steel	6.3056	45.1	40	20	39.94827982
29	window_glass015	0.004	2230	1.12	830	0.26	0.00104	glass	6.3056	0	40	20	40
30	air_wents004	0.03	2700	218	900	0.26	0.0078	steel	6.3056	71.2	40	20	39.90201835
31	air_wents005	0.03	2700	218	900	0.26	0.0078	steel	6.3056	71.2	40	20	39.90201835
32	air_wents006	0.03	2700	218	900	0.26	0.0078	steel	6.3056	71.2	40	20	39.90201835
33	air_wents007	0.03	2700	218	900	0.26	0.0078	steel	6.3056	71.2	40	20	39.90201835
34	Steel_door001	0.03	7800	50	550	1.5	0.045	metal	6.3056	61.8	40	20	39.6292
35	ladder006	0.035	2700	200	900	0.2	0.00175	steel	6.3056	94.5	40	20	39.834625
36	ladder007	0.035	2700	200	900	0.2	0.00175	steel	6.3056	97.9	40	20	39.828675

8.4 Validation Against Existing Models

To ensure reliability, outputs were compared with:

- **ENVI-met validation datasets** (Voelker et al. [10]).
- **EnergyPlus reference models** [11].
- **Satellite LST values from Sentinel-2 analysis** (Thong et al. [4]).

Findings:

- Roof temperatures matched **ENVI-met ranges (±2.5 °C)**.
- Wall surface patterns aligned with **EnergyPlus predicted heat flux trends**.
- SunCalc exposure matched **Sentinel-2 urban temperature hotspots** within **5–8% error margin**

8.5 System-Level Testing

8.5.1 Responsiveness & Latency

- CSV generation: ~1.2 sec.
- MATLAB execution: 7–9 sec (per simulation).
- End-to-end pipeline: ~12 sec average.

8.5.2 Stability

- Tested with 10 simultaneous simulation requests.
- No crashes; ~18% increase in latency.

8.5.3 Usability Feedback

- Test users found dashboard intuitive.
- Visual overlays (heatmaps + graphs) improved interpretability.

8.6 Interpretation of Results

- **Material Influence:** Concrete retains heat longest, while wood cools fastest.
- **Weather Sensitivity:** High humidity + low wind reduces cooling, worsening UHI.
- **Validation Success:** Results aligned with established models, proving reliability.
- **System Responsiveness:** End-to-end latency within acceptable real-time boundaries (<15 sec).

8.7 Key Findings

- The tool successfully integrates **GIS, 3D models, real-time weather, and MATLAB simulations** into a single pipeline.
- Simulation outputs were consistent with **ENVI-met and EnergyPlus models**, with small error margins.
- The **frontend visualization** provided heatmaps and graphs that were easily interpretable by test users.
- The **SunCalc integration** effectively captured diurnal variations in solar exposure.
- The **backend pipeline** demonstrated stable and efficient performance, supporting multiple requests.
- Limitations include dependency on **MATLAB runtime** (execution speed) and **accuracy of weather API inputs**.

This chapter presented the **experiments, results, and validation** of the UHI simulation tool. Multiple scenarios across different materials and weather conditions were tested. Results demonstrated realistic temperature distributions, exposure patterns, and alignment with validated models. The system proved stable, responsive, and effective in visualizing UHI dynamics.
The findings confirm that the tool is not only technically robust but also has strong potential as a **decision-support platform** for urban planners, architects, and environmental researchers.

9 Commercialization & Future Work

The proposed Urban Heat Island (UHI) simulation tool is envisioned to be commercialized as a fully integrated digital platform designed to support smart city planning, environmental sustainability initiatives, and urban infrastructure optimization. Unlike traditional academic prototypes that remain confined to research labs, this platform has been conceptualized with a strong commercialization pathway that ensures both accessibility and long-term scalability. The commercialization approach builds on the system's ability to merge 3D digital twin modeling, real-time weather integration, and MATLAB Simscape simulations into a single pipeline that delivers actionable insights for architects, engineers, urban planners, and policymakers.

The commercial product would be developed as a cloud-hosted platform accessible via a web interface and API endpoints, ensuring that stakeholders across different sectors can integrate it into their decision-making workflows. Users such as municipal authorities, environmental consultants, and construction companies could upload GIS and building models directly to the system, simulate temperature behavior under different climate conditions, and extract detailed reports. These reports could then be used to evaluate the effectiveness of interventions such as reflective roofing, façade greening, tree planting, or material substitutions. By offering both visual heatmaps and quantitative temperature outputs, the system provides dual value: it communicates findings effectively to policymakers and citizens through visuals, while also supplying engineers with precise, data-driven insights.

From a commercialization standpoint, a tiered subscription model would be applied to balance accessibility with financial sustainability. In its basic tier, the platform would allow users to run simulations with limited complexity, for example, single-building analysis with standard weather data. This ensures broad accessibility for universities, small research projects, and local councils operating with limited budgets. The premium tier, however, would unlock advanced features such as large-scale neighborhood simulations, high-resolution GIS overlays, AI-enhanced prediction models, and integration with third-party platforms like ENVI-met or EnergyPlus. Premium users would also gain access to historical weather datasets, batch processing of multiple building models, and extended visualization dashboards that support predictive urban planning at scale. This commercialization strategy allows the platform to serve both small organizations with specific needs and large-scale smart city projects requiring complex scenario modeling.

The application of this tool in smart cities extends beyond traditional UHI analysis. Smart cities require integrated digital solutions that can adapt to dynamic environmental conditions, and this platform has the potential to act as a digital backbone for urban resilience planning. For instance, municipalities could simulate how different zoning policies impact heat accumulation or assess how changes in building density and material choice influence neighborhood temperatures. In addition, the integration of real-time weather APIs allows the tool to serve as an early-warning system for heatwave conditions by modeling projected thermal stress across urban districts. This would enable proactive interventions such as temporary cooling shelters, traffic rerouting to reduce localized heat, or optimized irrigation schedules for urban greenery. As smart cities increasingly rely on digital twins for real-time monitoring, the proposed tool becomes a natural extension of that ecosystem.

Policy relevance is another major driver of commercialization. Global and regional climate policies, such as the Paris Agreement commitments and national climate adaptation strategies, emphasize the reduction of urban heat stress as a key resilience measure. However, policymakers often face the challenge of translating broad climate goals into localized actions. This tool directly addresses that gap by providing a science-backed, simulation-driven approach to policy evaluation. Governments could use the platform to test the effectiveness of proposed building codes, evaluate compliance with sustainability standards, and quantify the benefits of heat mitigation strategies such as cool pavements or rooftop gardens. By making simulation outputs easy to interpret and integrate into urban planning documents, the tool supports evidence-based policymaking while fostering transparency and accountability.

In terms of scalability, the system has been designed with modularity in mind. Each component—GIS import, Blender-based 3D modeling, SunCalc sunlight simulation, and MATLAB thermal analysis—can be independently upgraded or replaced with more advanced modules as technology evolves. In the near future, AI/ML approaches could be integrated to accelerate prediction accuracy by training models on large-scale simulation data. For example, once enough simulations have been conducted across multiple cities, a machine learning model could be trained to predict heat distribution without running full-scale MATLAB simulations, thereby reducing computational time and costs. Furthermore, the use of containerized microservices ensures that the platform can be deployed across different cloud environments, supporting scalability from single-city pilots to nationwide smart city programs.

The future work of this research will therefore focus on expanding both the technical capabilities and the application scope of the platform. On the technical side, additional parameters such as humidity, wind turbulence, and radiation balance could be incorporated for more holistic microclimate simulations. Integration with IoT sensors placed in real urban areas could also enhance real-time accuracy by continuously feeding the system with live temperature, air quality, and wind data. On the application side, the tool could evolve into a full-scale urban climate management platform by adding modules for air pollution dispersion modeling, energy consumption prediction, and climate adaptation strategy evaluation. Partnerships with city governments and international organizations would further drive adoption, positioning the tool as a central asset in global efforts to combat the adverse impacts of climate change in urban areas.

In summary, the commercialization and future work pathways of the proposed UHI simulation tool are strongly aligned with the pressing needs of modern urban development. By combining accessibility through tiered subscriptions, policy relevance through simulation-based evidence, and scalability via modular design, the platform offers a viable route toward becoming a cornerstone technology in smart city ecosystems. Its evolution into a real-time, AI-enhanced, and IoT-integrated digital twin platform will further expand its value proposition, enabling cities not only to analyze heat risks but also to proactively manage and mitigate them. This dual focus on immediate usability and long-term innovation ensures that the research extends beyond academic boundaries and contributes tangibly to sustainable, climate-resilient urban futures.

10 Budget & Timeline

The budget for the proposed Urban Heat Island (UHI) simulation platform has been carefully planned to ensure sustainable deployment, reliable operation, and scalability as adoption grows. The platform integrates multiple technologies, including a React + Three.js frontend for visualization, a Node.js backend for data handling, MATLAB Simscape for thermal simulation, and cloud-based infrastructure for hosting services. To achieve high performance and continuous availability, a hybrid budget strategy is adopted, combining one-time development and deployment costs with recurring operational expenses.

The major cost components include cloud infrastructure for backend and simulation hosting, storage and database solutions, software deployment across platforms, and ongoing support and monitoring. Additional considerations have been given to secure communication channels, licensing requirements, and contingency allocations to ensure uninterrupted service delivery.

Table 10-1 Budget plan

ITEM	COST (LKR)	FREQUENCY	JUSTIFICATION
Cloud Server (Backend + MATLAB Pipeline)	12,000	Monthly	VPS/cloud instance to handle Node.js backend, MATLAB integration, and simulation processing efficiently.
Web Dashboard Hosting (React + Three.js)	2,500	Monthly	Hosting for simulation dashboard and visualization interface.
Domain + SSL Certificate	2,000	Monthly	Enables secure HTTPS access and branded access points for the system.

MATLAB Academic/Research License	15,000	Monthly	License required for Simscape Thermal simulations and integration with pipeline.
Monitoring & Logging Tools	1,500	Monthly	For monitoring up time, simulation performance, and logging anomalies.
Email & Notification Services	1,200	Monthly	For sending alerts, simulation completion notifications, and scheduled updates.
Support & Maintenance Allocation	6,000	Monthly	Developer allocation for patching, updates, and responding to user feedback.
Contingency Allocation	10,000	Ont-Time	Reserved for unexpected costs in scaling or integration during deployment.

• **Total Monthly Operational Cost: ~LKR 28,700 – 30,000**

• **Total Annual Licensing Cost: ~LKR 15,000**

• **Total One-Time Cost: ~LKR 10,000**

10.1 Budget Justification Summary

- **Cloud Server:** Required to run backend services, Node.js API endpoints, and MATLAB pipeline integrations. Ensures simulations execute with minimal latency and scale dynamically with workloads.
- **Database Hosting:** Essential for managing simulation inputs and outputs, including GIS metadata, building properties, and thermal simulation logs.
- **Object Storage:** Needed for handling large 3D model files, snapshot images, and CSV-based simulation outputs in a scalable and secure manner.
- **Web & Dashboard Hosting:** Provides the core interface for users to upload models, run simulations, and visualize results.
- **MATLAB License:** Enables the thermal modeling engine via Simscape. While academic/research licenses reduce costs, a dedicated allocation ensures long-term accessibility.
- **Monitoring Tools:** Provide operational transparency by tracking uptime and performance, ensuring that simulations remain reliable under different load conditions.
- **Email & Notification Services:** Enhance usability by keeping users informed of simulation progress and completion.
- **Support & Maintenance:** Ensures that the system can be continually improved with updates, bug fixes, and feature additions.
- **Contingency Fund:** Reserved to address unforeseen technical or operational issues, providing flexibility during rollout.

10.2 Project Timeline

The project timeline is designed across four key phases over **12 months**, ensuring structured development, deployment, and evaluation.

Table 10-2 Project timeline

Phase	Duration	Key Activities
Phase 1 – Initial Development	Months 1–3	GIS import pipeline setup, 3D modeling integration with Blender, metadata mapping, backend (Node.js) prototype.
Phase 2 – Core Simulation Integration	Months 4–6	MATLAB Simscape pipeline integration, SunCalc sunlight exposure module, CSV handling, API communication.
Phase 3 – Frontend & Visualization	Months 7–9	React + Three.js dashboard development, interactive visualization, snapshot capture, report generation.
Phase 4 – Testing & Deployment	Months 10–12	System validation (against ENVI-met/EnergyPlus), user testing, cloud deployment, monitoring setup, documentation.

The timeline ensures that core components are built sequentially but with overlapping iterations to allow early testing. For example, while Phase 2 handles MATLAB integration, Phase 3 visualization can already begin using simulated data stubs. This parallel workflow ensures efficiency and early bug detection. By the final quarter, the system is fully validated, tested, and deployed for commercial or academic use.

11 GANTT CHART

Figure 11-1 Gantt Chart

12 CONCLUTION

Through careful design, development, and evaluation, the proposed Urban Heat Island (UHI) simulation tool has demonstrated its capacity to provide meaningful insights into the thermal behavior of buildings and urban environments under diverse environmental conditions. By integrating GIS data, digital twin modeling in Blender, weather API inputs, SunCalc-based solar exposure analysis, and MATLAB Simscape Thermal simulations, the system successfully offers a comprehensive framework for predicting and visualizing temperature variations across different building components and material configurations.

The project outcomes confirm that the pipeline not only produces results consistent with established platforms such as ENVI-met and EnergyPlus but also delivers them in a manner that is lightweight, adaptable, and user-friendly. Validation experiments showed that predicted surface temperatures remained within an acceptable error margin, reinforcing the reliability of the approach. The inclusion of detailed building metadata—such as material thickness, conductivity, heat capacity, and mass—further enhanced simulation fidelity and allowed for results that closely reflect real-world thermal dynamics.

Importantly, the system has been designed to prioritize usability and accessibility. Its React and Three.js frontend enables users to interact with 3D models, capture snapshots, and view simulation outputs visually, reducing the technical barrier often associated with advanced modeling tools. This intuitive design ensures that the platform can be adopted not only by researchers but also by urban planners, architects, and policymakers seeking evidence-based strategies to mitigate UHI effects.

The commercialization potential of the system is equally promising. As cities worldwide grapple with rising urban temperatures and the associated challenges of energy demand, human comfort, and sustainability, the platform can be positioned as a practical decision-support tool for smart city initiatives. Its scalable architecture, designed for integration with weather data sources and capable of cloud deployment, ensures that it can expand in scope to accommodate larger datasets, multiple users, and real-time analytics. Such scalability makes the solution viable for deployment in municipal planning offices, research institutions, and consulting firms working in climate adaptation and energy-efficient urban design.

Overall, this research contributes meaningfully to the advancement of UHI modeling by bridging the gap between high-fidelity scientific simulations and accessible, user-oriented software solutions. It demonstrates that with thoughtful integration of digital twin technologies, environmental data, and thermal modeling, it is possible to deliver actionable insights that support climate-resilient urban planning. With continued refinement, wider validation across global cities, and potential integration with IoT sensors and AI-driven predictive models, the system has the potential to become a transformative tool in addressing the challenges of urban heat and supporting the creation of more sustainable, livable cities.

13 REFERENCES

[1] Y. K. H. C. S. J. R. S. W. J. Kyungil Lee, "Trend Analysis of Urban Heat Island Intensity According to Urban Area Change in Asian Mega Cities," *Sustainability,* vol. 12, no. 1, p. 11, 2020.

[2] E. J. G. H. a. M. J. Kim, "Urban heat in South Asia: Integrating people and place in adapting to rising temperatures," Policy Brief. Washington DC. The World Bank., 2023. [Online]. Available: https://weadapt.org/knowledge-base/cities-and-climate-change/urban-heat-in-south-asia-integrating-people-and-place-in-adapting-to-rising-temperatures/. [Accessed 2024].

[3] N. A. N. A. A. Y. H. M. S. A. E.-S. Ali Najah Ahmed, "The urban heat Island effect: A review on predictive approaches using artificial intelligence models," *City and Environment Interactions,* vol. 28, p. 22, 2024.

[4] D. &. C. H. &. P. T. &. L. N. &. N. T. &. X. T. &. Q. T. Thong, "Analysis of urban heat islands combining Sentinel 2 and Landsat 8 satellite images in Hochiminh city," *Earth and Environmental Science.,* vol. 1349, p. 012032, 2024.

[5] S. S. Justyna Dębicka, "Comparative analysis of arcGISs and qGIS in terms of the transformations' runtime," *Geoinformatica Polonica,* vol. 17, pp. 99-108, 2018.

[6] W. L. a. O. D. a. J. A. a. K. R. a. J. C. a. G. L. a. P. K. a. T. G. a. L. Ahuja, "Environmental modeling framework invasiveness: Analysis and implications," *Environmental Modelling & Software,* vol. 26, no. 1364-8152, pp. 1240-1250, 2011.

[7] A. H. a. A. A. a. A. S. K. a. I. W. a. A. A. A. a. A. S. Pillai, "A comprehensive review of Digital Twin technologies in smart cities," *Digital Engineering,* vol. 4, no. 2950-550X, p. 100040, 2025.

[8] P. Beshai, "3D Data Visualization with React and Three.js," 31 01 2020. [Online]. Available: https://medium.com/cortico/3d-data-visualization-with-react-and-three-js-7272fb6de432. [Accessed 20 04 2025].

[9] S.-J. M. a. J. H. a. Y. F. a. C. Y. a. Y. Xue, "CFD simulations on the wind and thermal environment in urban areas with complex terrain under calm conditions," *Sustainable Cities and Society,* vol. 118, p. 106022, 2025.

[10] H. A. a. M. H. a. R. H. a. C. Voelker, "ENVI-met validation data accompanied with simulation data of

the impact of facade greening on the urban microclimate," *Data in Brief,* vol. 42, p. 108200, 2022.

[11] U. D. o. Energy, "EnergyPlus," 29 03 2022. [Online]. Available: https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v22.1.0/EngineeringReference.pdf. [Accessed 30 03 2025].

[12] I. The MathWorks, "Thermal Models," The MathWorks, Inc., 2022. [Online]. Available: https://www.mathworks.com/help/simscape/thermal-models.html. [Accessed 12 04 2025].

[13] M. H. S. Y. C. W. Siavash Ghorbany, "Towards a Sustainable Urban Future: A Comprehensive Review of Urban Heat Island Research Technologies and Machine Learning Approaches," *Sustainability,* vol. 16, no. 11, 2024.

[14] W. API, "Weather API," OPLAO FZCO, [Online]. Available: https://www.weatherapi.com/. [Accessed 25 04 2025].

[15] A. M. M. a. C. H.-W. a. S. M. M. F. a. W. C. Y. a. M. N. a. J. E. Irfeey, "Sustainable Mitigation Strategies for Urban Heat Island Effects in Urban Areas," *Sustainability,* vol. 15, no. 2071-1050, 2023.

[16] M. V. a. P. D. a. K. E. a. V. W. a. L. K. a. P. Kluson, "Disposable indicator card for personal monitoring of solar exposure," *Journal of Photochemistry and Photobiology A: Chemistry,* vol. 454, no. 1010-6030, p. 115741, 2024.