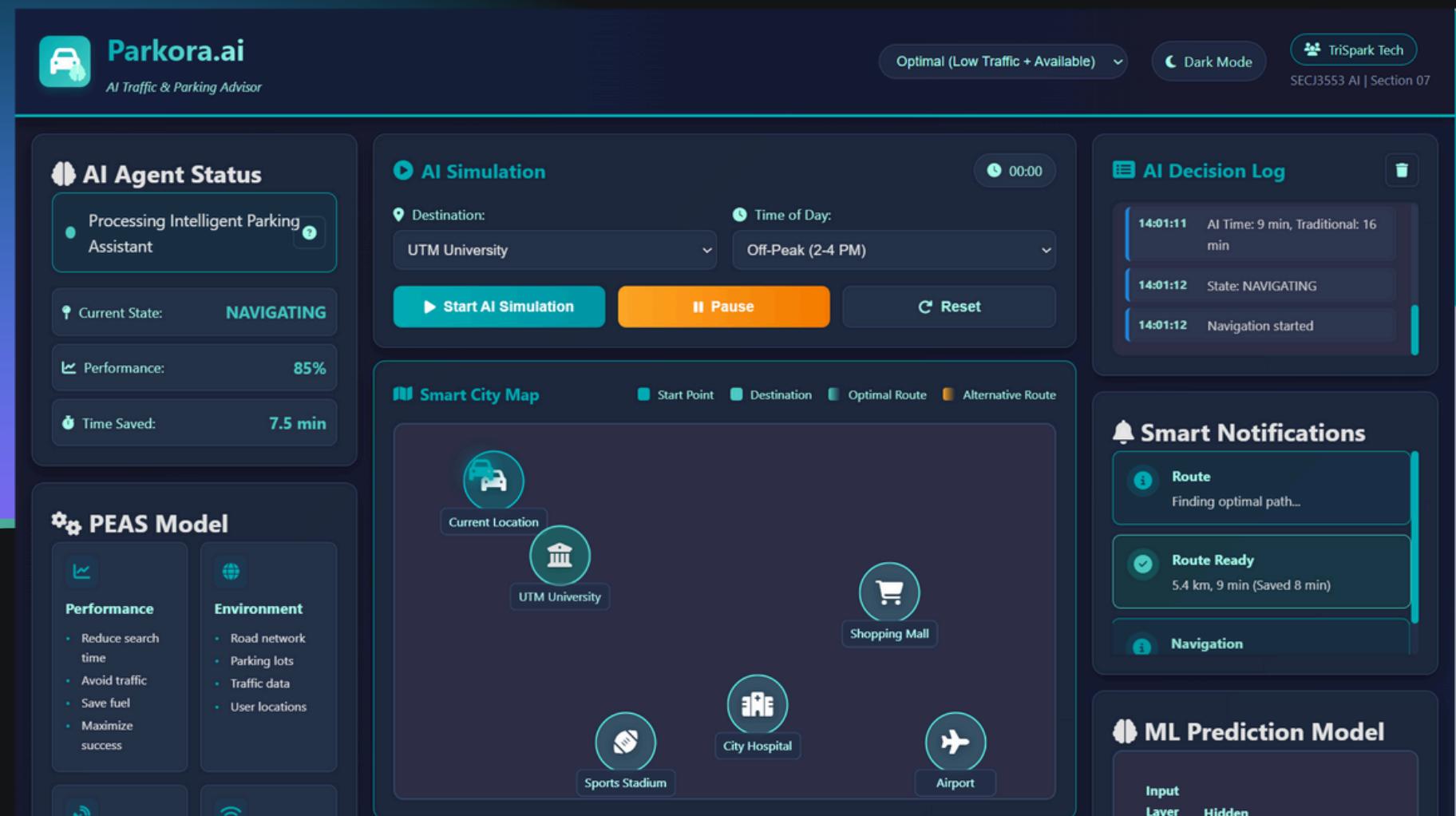


TRISPARK TECH

Parkora.ai



SECJ3553 Artificial Intelligence
Section 07

TASK 5: PROTOTYPE DEVELOPMENT AND
PROOF OF CONCEPT (POC)

Project Theme:
Smart City - Traffic & Parking Advisor Application
(Parkora.ai)

Our Team



YAP EN THONG



**AUSTIN SEE YONG
HUI**



WONG JIA XUAN



The Problem

- Research shows up to 30% of urban traffic
- Drivers search for parking
- Wastes fuel, increase pollution, stresses everyone out



Our Solution Parkora.ai

Intelligent parking advisor that uses artificial intelligence to guide drivers to available parking in real-time.



Welcome to Parkora.ai

AI-Powered Traffic & Parking Advisor for Smart Cities

Experience intelligent parking solutions using state-of-the-art AI technologies including First Order Logic, State Space Search, and Machine Learning predictions.



30%

Time Saved



25%

Fuel Saved



88%

Satisfaction



92%

Success Rate

Quick Start Simulation

Select Destination



UTM University



Shopping Mall



City Hospital



Airport



Sports Stadium

Strategic Vision & POC Approach



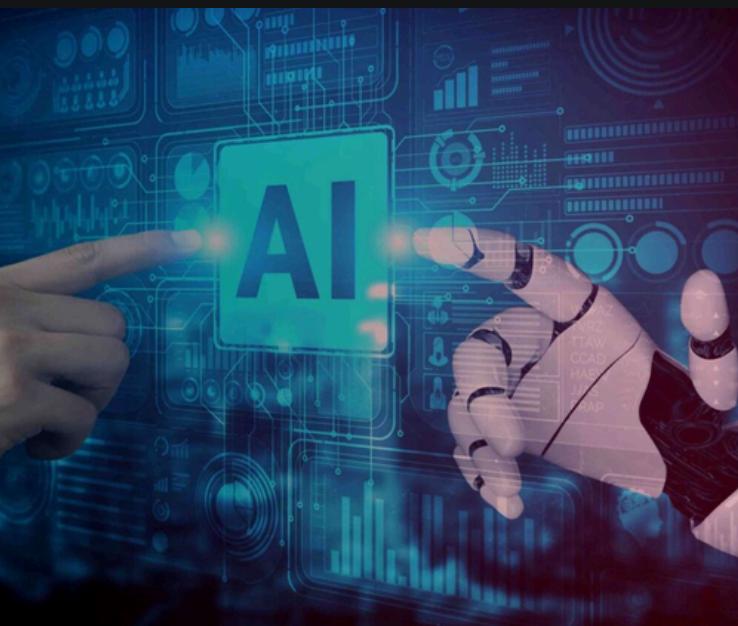
1. Design the experience with Figma prototypes to ensure it's user-friendly



```
index.html
...
<!DOCTYPE html>
<html lang="en" data-theme="dark">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <title>Parkora.ai - AI Traffic & Parking Advisor</title>
  <link rel="stylesheet" href="styles.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css">
  <script src="https://d3js.org/d3.v7.min.js"></script>
</head>
<body>
  <div class="container">
    <!-- Header -->
    <header class="header">
      <div class="logo">
        <div class="logo-icon">
          <i class="fas fa-car"></i>
          <i class="fas fa-brain"></i>
        </div>
        <div class="logo-text">
          <h1>Parkora.ai</h1>
          <span class="tagline">AI Traffic & Parking Advisor</span>
        </div>
      </div>
    </header>
  </div>
</body>

```

2. Build a fully functional web-based POC that simulates a smart city environment



3. AI implementation

Parkora.ai AI Traffic & Parking Advisor

Welcome to Parkora.ai

AI-Powered Traffic & Parking Advisor for Smart Cities

Experience intelligent parking solutions using state-of-the-art AI technologies including First Order Logic, State Space Search, and Machine Learning predictions.

AI Recommended

#1 UTM Main Parking
UTM University
Distance: 8.5 km, Travel Time: 12 min, Cost per Hour: \$2.5/hr
AI Score: 95%
Walking: 5 min

#2 UTM Alternative Lot
UTM University
Distance: 9.2 km, Travel Time: 15 min, Cost per Hour: \$2/hr
AI Score: 78%

#3 Nearby Street Parking
Near UTM
Distance: 7.8 km, Travel Time: 11 min, Cost per Hour: \$1.5/hr
AI Score: 52%

AI Reasoning

- Optimal route with low congestion (Traffic Factor: 0.3)
- High availability probability (92% confidence)
- Best cost-to-convenience ratio
- Historical data shows consistent availability at this time

AI Score: 95%

Parkora.ai AI Traffic & Parking Advisor

AI Agent Status
Processing Intelligent Parking Assistant

Current State: **PARKING_REACHED**

Performance: 85%

Time Saved: 5 min

PEAS Model

- Performance: Reduce search
- Environment: Road network

Simulation Controls
Start, Pause, Reset

Smart City Map
Current Location, UTM University, Shopping Mall, City Hospital, Airplane icon

AI Decision Log

- 18:43:21 State: ROUTE_CALC
- 18:43:21 Calculating optimal route with Dijkstra algorithm...
- 18:43:23 State: ROUTE_READY

Performance Metrics

Metric	Value
Time Saved	5 min
Fuel Saved	0.40 L
Efficiency	85%
Success Rate	92%

FIGMA

- Map out the complete user journey
- Design intuitive interfaces for homepage, map views, and notifications
- Validate our user flow before development
- Collaborate effectively as a team"

Working Prototype

- Live Demo

- Frontend: HTML5, CSS3, JavaScript – platform-independent, zero-dependency deployment
- Visualisation: D3.js – enables interactive state space and algorithm animations
- Algorithms: Dijkstra/A* – classical search algorithms implemented in pure JavaScript
- Simulation: Mock data system – models urban dynamics without external API dependencies

Functional Validation

Real-Time Adaptation Verified

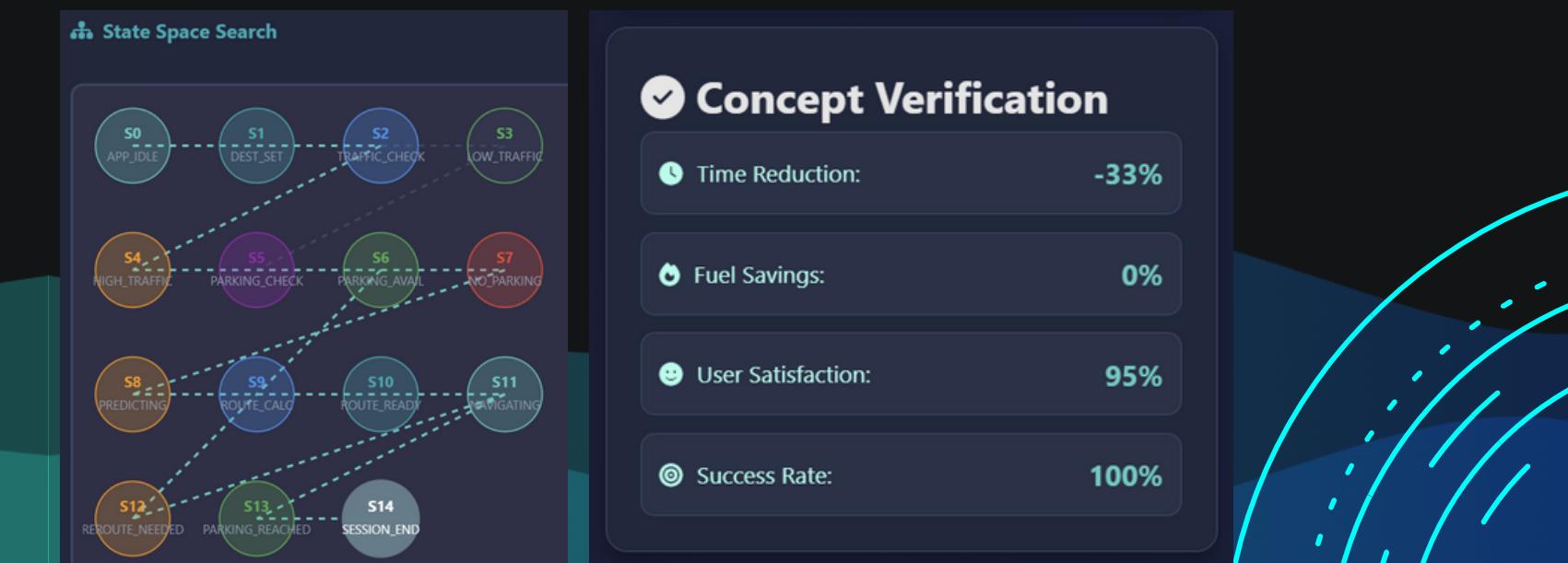
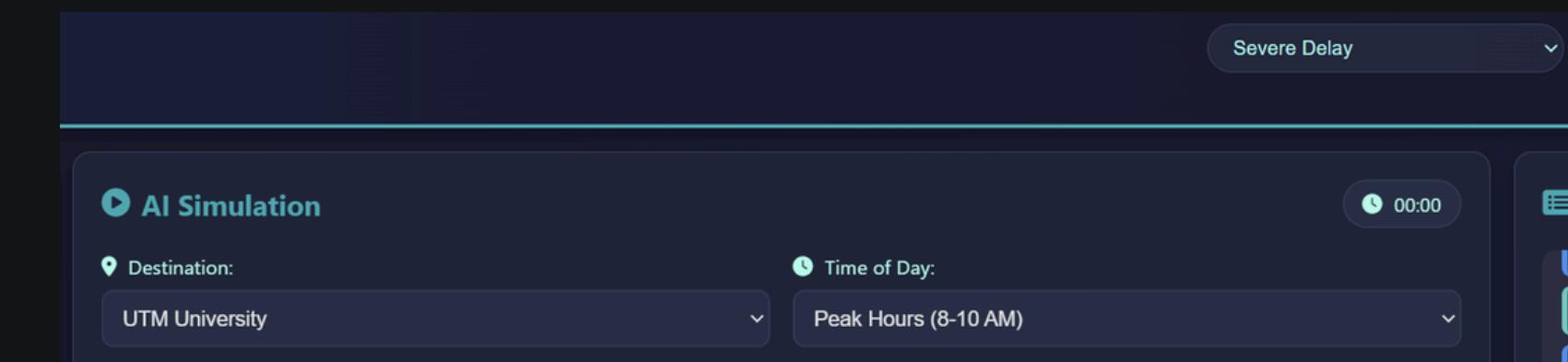
- Our system adapts to different urban conditions.
- When we switch to “Severe Delay”, the AI immediately recalculates, showing real-time decision making.

Personalized Decisions

- User can select destination and time of day. User's preferences change the result of the AI calculation.

Quantifiable, Real-time Metrics

- Every simulation generates live metrics, such as time reduction and fuel savings which are from actual route algorithms.



Technical Verification

FOL Knowledge Representation

- From AI Decision Explanation, it transforms 5 FOL rules into working logic.

$$\begin{aligned} \forall p (\text{ParkingLot}(p) \rightarrow (\text{Available}(p) \leftrightarrow \exists s (\text{Space}(s) \wedge \text{In}(s, p) \wedge \neg \text{Occupied}(s)))) \\ \forall r \forall p ((\text{Route}(r) \wedge \text{Connects}(r, \text{UserLocation}, p) \wedge \text{Available}(p) \wedge \text{LeastCost}(r)) \rightarrow \text{OptimalRoute}(r)) \\ \forall p \forall t ((\text{ParkingLot}(p) \wedge \text{HighOccupancy}(p, \text{time_past})) \rightarrow \text{LikelyUnavailable}(p, \text{time_now})) \\ \forall p ((\text{Available}(p) \wedge \text{Near}(p, \text{Destination})) \rightarrow \text{Preferred}(p)) \\ \forall p ((\text{Available}(p) \wedge \text{Near}(p, \text{Destination})) \rightarrow \text{Preferred}(p)) \end{aligned}$$

State Space Search Visualization

- Each node represents an actual state in our system.
- Real-time transition tracking is visualized.

Dijkstra's Algorithm Live Demo

- Green nodes are visited, yellow are in the frontier and the path shows in real-time.

Complete PEAS Model

- Every component is implemented. Performance measures update live; Environment conditions change with scenarios; Actuators triggers notifications and Sensors simulate real-world data.



Quantifiable Impact

In optimal situation,

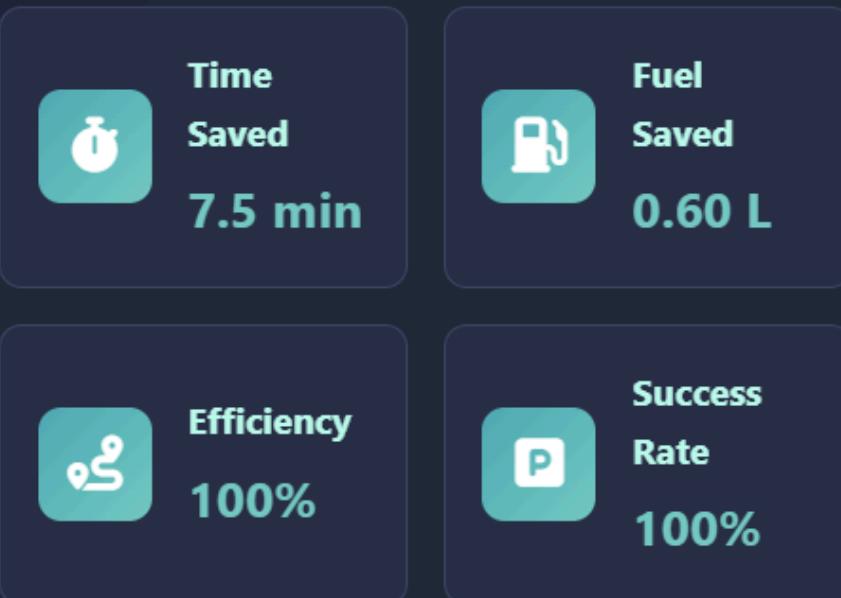
- Time saved: 7.5 min per parking search
- Fuel saved: 0.60L
- Success Rate: 100% on the first attempt to find an available spot.

The traditional success rate (53%) is lower compared to AI-assisted success rate (100%).

This data validates our core value proposition: Parkora.ai reduces urban friction, saves drivers time and money and contributes to lower emissions by cutting down unnecessary traffic.

The screenshot shows the 'AI Simulation' section of the app. It includes fields for 'Destination' (UTM University), 'Time of Day' (Off-Peak (2-4 PM)), and a dropdown for 'Optimal (Low Traffic + Available)' settings. There are also icons for a clock (00:00) and a list.

Performance Metrics



Traditional vs AI Performance

Traditional	AI-Assisted
53%	100%

Concept Verification

Time Reduction:	47%
Fuel Savings:	6%
User Satisfaction:	95%
Success Rate:	100%

Practical Relevance and Future Vision

Scalable Architecture

Our system is designed for integration. The PEAS model shows exactly where IoT sensors and traffic APIs would connect.

Smart City Ready

It's a functional interface which can change the destination, simulating how real users would interact.

Educational Value Proved

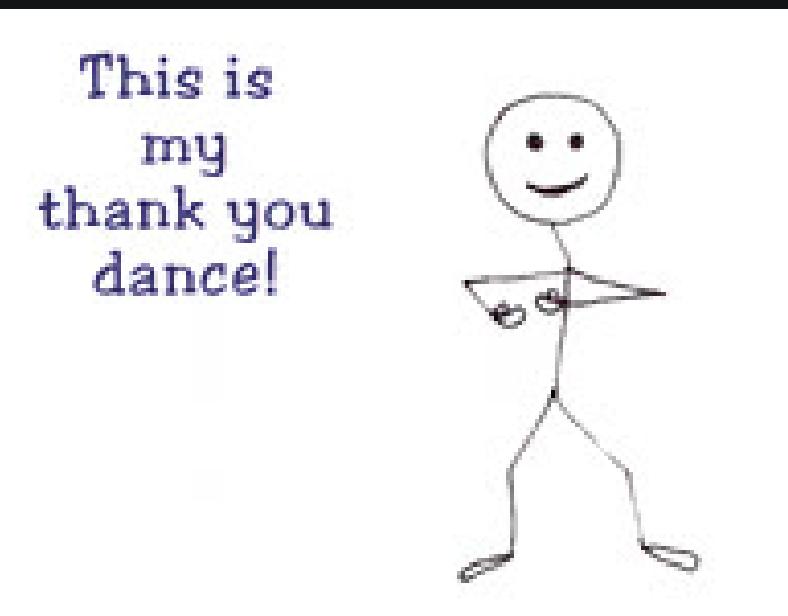
We have successfully transformed concepts from FOL to state space search which is a working system that solves actual urban problems.

Summary

- 1 **Problem Solved:**
Urban parking inefficiency.
- 2 **Solution Built:**
AI-powered parking advisor
- 3 **Technology Used:**
FOL, State Space Search, PEAS, ML
- 4 **Impact Measured:**
Measurable improvements proven through testing



Thank You



QNA SESSION