



Fair Distance Pricing: Implementing a Toll Tax System Based on Distance

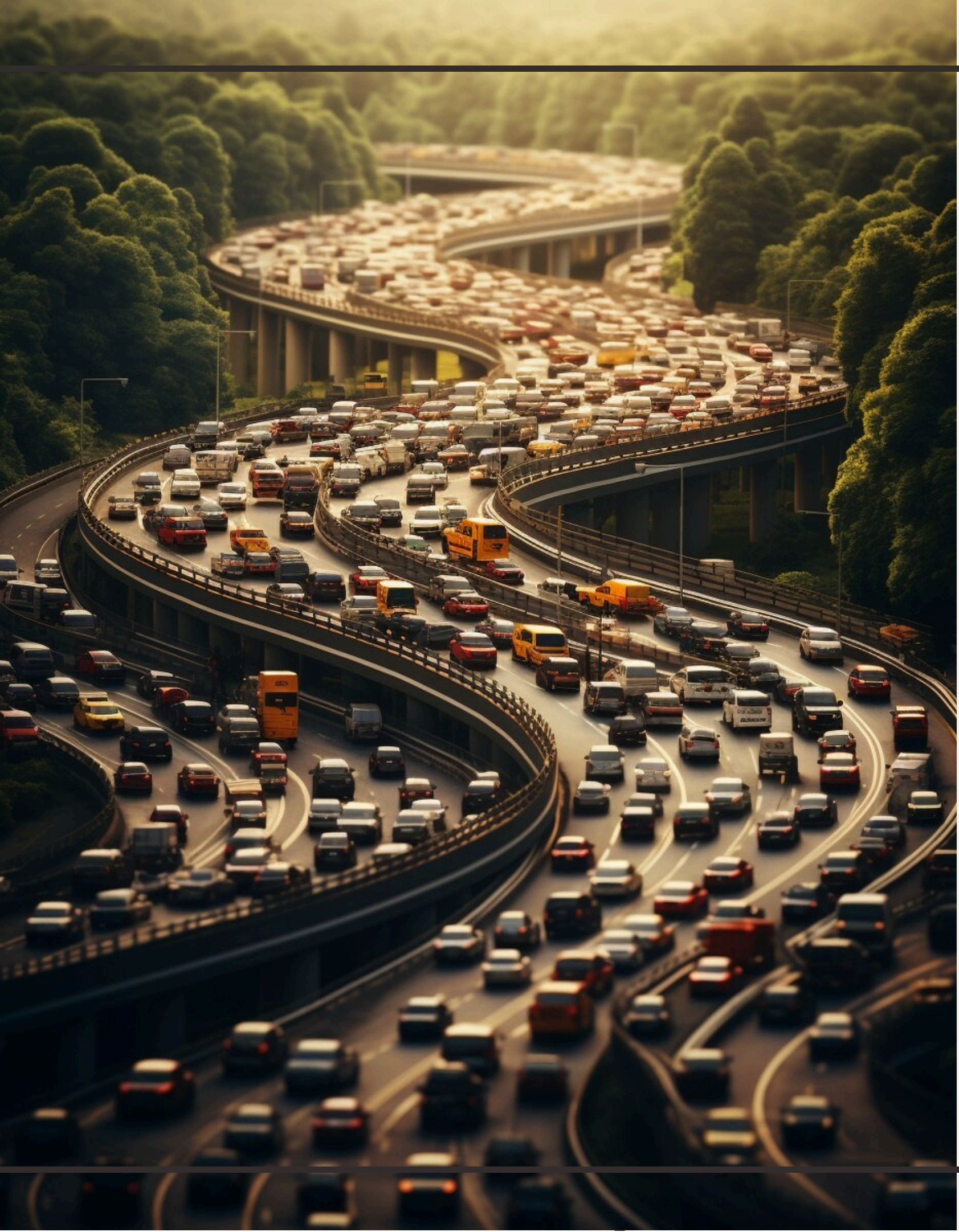
Fair Distance Pricing aims to create an equitable **toll tax system** based on actual distance. The current toll tax system in India levies a fixed fee on all vehicles passing through a toll plaza, regardless of the distance traveled. This leads to inequity for short-distance travelers who end up paying disproportionately high tolls.



What is Distance-Based Tolling?



Distance-Based Tolling is a system where drivers pay fees based on the distance they travel. This method encourages **efficient driving** habits and can reduce traffic congestion. By charging based on actual usage, it ensures that road maintenance costs are fairly distributed among users.



Benefits of Distance Pricing

Implementing a **toll tax system** based on mileage offers several advantages. It promotes **fairness** by ensuring that those who use the roads more contribute more. Additionally, it can lead to reduced traffic congestion, lower emissions, and better funding for road maintenance and infrastructure improvements.

Slide1: Registration

- Users download a dedicated mobile app. The app collects user and vehicle information (license plate, vehicle type, etc.). It enables users to link their payment methods. The app acts as an interface for toll payments and provides trip history.
- Central Server:
- Stores vehicle and user data.
- Processes GPS data from smartphones.
- Calculates distance traveled and toll amount.
- Integrates with payment gateways.
- Manages user accounts and payment history.
- Provides data analytics for traffic patterns and revenue optimization.

```
// User Registration (React)
const registerUser = async (userData) => {
  const response = await axios.post('/api/auth/register', userData);
  return response.data;
};

// Trip History (React)
const fetchTrips = async () => {
  const response = await axios.get('/api/trips');
  setTrips(response.data);
};
```

Slide2: Central Server and GPS Data

Central Server:

Stores vehicle and user data.

Processes GPS data from smartphones.

Calculates distance traveled and toll amount.

Integrates with payment gateways.

Manages user accounts and payment history.

Provides data analytics for traffic patterns and revenue optimization.

```
// GPS Tracking (React Native)
import { useEffect } from 'react';
import Geolocation from '@react-native-community/geolocation';

useEffect(() => {
  Geolocation.getCurrentPosition(
    (position) => {
      console.log("Latitude: ", position.coords.latitude);
      console.log("Longitude: ", position.coords.longitude);
    },
    (error) => console.log(error),
    { enableHighAccuracy: true, timeout: 20000, maximumAge: 1000 }
  );
}, []);
```

```
// Server Setup (Node.js)
const express = require('express');
const mongoose = require('mongoose');
const app = express();

mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true, useUnifiedTopology: true });

app.use(express.json());
app.use('/api/auth', require('./routes/authRoutes'));
app.use('/api/trips', require('./routes/tripRoutes'));
app.listen(5000, () => console.log('Server running on port 5000'));
```

Slide4: Toll Road Mapping and Distance Calculation, Payement and Receipts.

- The system calculates the distance traveled based on GPS coordinates and toll road maps.
- The toll amount is calculated based on the distance traveled, vehicle type, and time of day.
- The calculated toll is automatically deducted from the user's linked payment method.
- A digital receipt is generated and stored in the user's account.

```
// Distance Calculation (Node.js)
const calculateDistance = (startCoords, endCoords) => {
  const R = 6371; // Radius of the earth in km
  const dLat = deg2rad(endCoords.lat - startCoords.lat);
  const dLon = deg2rad(endCoords.lng - startCoords.lng);
  const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) + Math.cos(deg2rad(
    startCoords.lat)) * Math.cos(deg2rad(endCoords.lat)) * Math.sin(dLon / 2)
  * Math.sin(dLon / 2);
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  const distance = R * c; // Distance in km
  return distance;
};

// Toll Calculation (Node.js)
const calculateToll = (distance, vehicleType) => {
  const ratePerKm = vehicleType === 'car' ? 0.5 : 0.8; // Example rates
  return distance * ratePerKm;
};
```

```
// Payment Processing (Node.js with Stripe)
const stripe = require('stripe')(process.env.STRIPE_SECRET_KEY);

exports.processPayment = async (req, res) => {
  const { amount, paymentMethodId } = req.body;
  try {
    const paymentIntent = await stripe.paymentIntents.create({
      amount,
      currency: 'usd',
      payment_method: paymentMethodId,
      confirm: true,
    });
    res.status(200).json({ success: true, paymentIntent });
  } catch (error) {
    res.status(500).json({ message: 'Payment processing failed' });
  }
};
```

Thanks!

Do you have any questions?
anujpandey0513@gmail.com
+91 9455080726

<https://github.com/AnujPandey123>
<https://www.linkedin.com/in/anuj-pandey-551bb6226/>

