INTERNET OF THINGS _PHASE-3

PUBLIC TRANSPORT OPTIMIZATION

PROJECT TITLE:

Building a public transport optimization system using dataset loading and preprocessing

INTRODUCTION:

Public transport optimization is a critical endeavor aimed at enhancing the efficiency, accessibility, and sustainability of urban transportation systems. As cities continue to grow and face congestion challenges, optimizing public transit networks becomes imperative to reduce traffic, improve air quality, and provide affordable and convenient mobility solutions for residents. This introduction will explore the key principles, strategies, and technologies involved in the optimization of public transport, highlighting its significance in shaping the future of urban mobility.

OBJECTIVES:

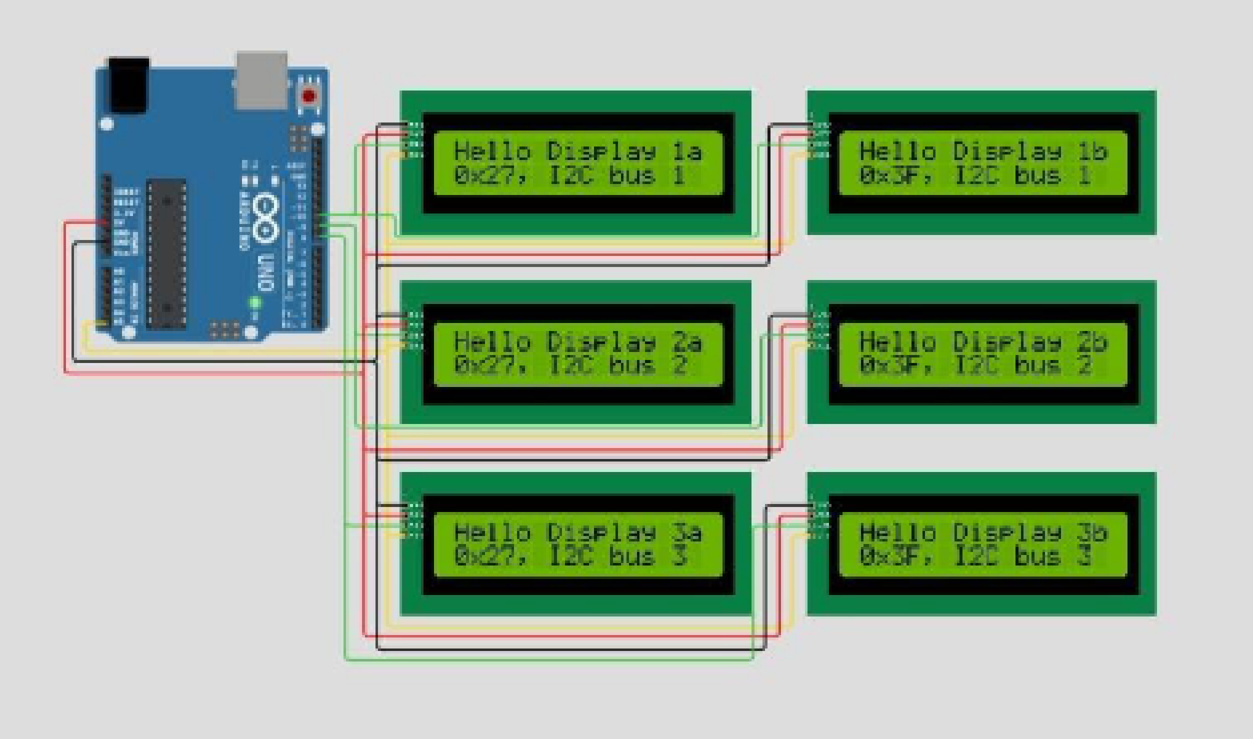
Objectives for public transport typically refer to the overarching goals and aims of a public transport these objectives can vary depending on the specific needs and priorities of a city or region, but they generally aim to provide efficient, sustainable, and accessible transportation options to the public system. These objectives can vary from one location to another, but they often include.

PROGRAM:

```
#include<Software Wire .h>

Software Wirewire1 (10,SCL);
Software Wirewire2 (9,SCL);
Software Wirewire3 (8,SCL);
```

```
#include "LiquidCrystal_I2C_soft.h"


LiquidCrystal_I2C lcd1a (&wire1, 0x27, 20, 4);
LiquidCrystal_I2C lcd1b (&wire1, 0x3F, 20, 4);
LiquidCrystal_I2C lcd2a (&wire2, 0x27, 20, 4);
LiquidCrystal_I2C lcd2b (&wire2, 0x3F, 20, 4);
LiquidCrystal_I2C lcd3a (&wire3, 0x27, 20, 4);
LiquidCrystal_I2C lcd3b (&wire3, 0x3F, 20, 4);

Void setup ()
{
 lcd1a.init ();
 lcd1a.backlight ();
 lcd1a.setCursor (0,0);
 lcd1a.print ("Hello Display 1a");
 lcd1a.setCursor (0,1);
 lcd1a.print ("0x27, I2C bus 1");

 lcd1b.init ();
 lcd1b.backlight ();
 lcd1b.setCursor (0, 0);
 lcd1b.print ("Hello Display 1b");
 lcd1b.setCursor (0, 1);
 lcd1b.print ("0x3F, I2C bus 1");

 lcd2a.init ();
 lcd2a.backlight ();
 lcd2a.setCursor (0,0);
 lcd2a.print ("Hello Display 2a");
 lcd2a.setCursor (0,1);
 lcd2a.print ("0x27, I2C bus 2");

 lcd2b.init ();
 lcd2b.backlight ();
 lcd2b.setCursor (0, 0);
 lcd2b.print ("Hello Display 2b");
 lcd2b.setCursor (0, 1);
 lcd2b.print ("0x3F, I2C bus 2");

 lcd3a.init ();
 lcd3a.backlight ();
 lcd3a.setCursor (0,0);
 lcd3a.print ("Hello Display 3a");
 lcd3a.setCursor (0,1);
 lcd3a.print ("0x27, I2C bus 3");

 lcd3b.init ();
 lcd3b.backlight ();
 lcd3b.setCursor (0, 0);
 lcd3b.print("Hello Display 3b");
 lcd3b.setCursor(0, 1);
 lcd3b.print("0x3F, I2C bus 3");
}
Void loop () {}
```

OUTPUT:



```
Hello Display 1a    Hello Display 1b
0x27, I2C bus 1     0x3F, I2C bus 1

Hello Display 2a    Hello Display 2b
0x27, I2C bus 2     0x3F, I2C bus 2

Hello Display 3a    Hello Display 3b
0x27, I2C bus 3     0x3F, I2C bus 3
```

STEPTS INVOLVED IN THIS PROGRAM:

1. Data Loading:

   You'll start by loading your dataset. In this example, we'll use a CSV file as a sample dataset. You can replace it with your own dataset.

python

```
import pandas as pd

# Load your dataset

Data set = p d . read  c s v ('public_transport_data.csv')
```

2. Data Preprocessing:

Data preprocessing is essential to clean, format, and prepare your dataset for optimization. Here are some common preprocessing steps: a. Handling Missing Data:

Python

```
# Remove rows with missing values or impute missing data

dataset .  Drop (place =True)
```

b. Data Transformation:

Python

```
# Convert time-related columns to date time objects

dataset['departure _time'] = pd. to_ date time(dataset['departure _ time'])
```

c.Feature Engineering:

You can create new features based on your dataset. For example, you can extract the day of the week, time of day, or calculate travel times.

Python

```
Dataset ['day_ of_ week'] = data set ['departure _ time']   . dt . Day of week

Dataset ['hour _ of_ day'] = data set ['departure _time'] .dt . hour
```

d. Filtering Data:

You might want to filter data based on specific criteria, such as routes, days, or stations.

Python

```
# Filter data for a specific route

dataset = dataset[dataset['route'] == 'Route1']
```

Data Exploration and Visualization:

2. It's important to visualize and explore your data to understand its characteristics.

Python

```
import  mat plot lib .p y plot as   pl t


# create plots or summary statistics to better understand the data

P l t. h I s t  (dataset ['passenger _count'])

P l. x label ('Passenger Count')

Pl t .y label ('Frequency')

p l t .show()
```

CONCLUSION:

Optimizing public transport is essential for creating more efficient sustainable, and accessible urban mobility systems .By investing in technology, infrastructure, and policy improvement, cities can reduce congestion, lower emissions, and enhance the overall quality of life for their 21st century and provide equitable transportation .