# PUBLIC TRANSPORT OPTIMIZATION

## PHASE 5 PROJECT

**Introduction:**

Optimizing public transport involves various aspects such as route planning, scheduling, and passenger information systems. It can be a complex task depending on the specific goals and constraints. Below, I'll provide a basic Python code example for optimizing bus routes using a genetic algorithm. This is a simplified illustration, and real-world public transport optimization projects are typically much more complex.

**Components required and system design:**

Optimizing public transport systems involves integrating various components and technologies to enhance efficiency, convenience, and sustainability. Here are some key components and strategies that can contribute to the optimization of public transport:

1**. Intelligent Transport Systems (ITS**):

Implementing ITS can improve the overall efficiency and safety of public transportation networks. This includes technologies such as real-time passenger information systems, traffic signal priority for buses, and automatic vehicle location systems.

2**. Integrated Ticketing and Payment Systems:**

Developing a seamless, integrated ticketing and payment system across different modes of public transport simplifies the process for passengers. This can include contactless payment methods, smart cards, and mobile ticketing applications.

3. **Fleet Management and Maintenance**:

Regular maintenance and effective management of public transport fleets are essential for ensuring operational efficiency and passenger safety. Implementing predictive maintenance technologies can help prevent breakdowns and improve overall fleet reliability.

4**. Infrastructure Development**:

Investing in well-planned infrastructure, such as dedicated bus lanes, priority signaling, and intermodal transit hubs, can significantly improve the speed and reliability of public transportation services.

5**. Data-Driven Decision Making:**

Leveraging data analytics and passenger feedback to make informed decisions about route planning, scheduling, and service adjustments can lead to better optimization of public transport operations.

6**. Eco-friendly Initiatives**:

Introducing environmentally friendly technologies and fuel-efficient vehicles, such as electric buses and hybrid trains, can contribute to reducing the carbon footprint of public transportation systems and promoting sustainable urban mobility.

7. **Accessibility and Inclusivity**:

Ensuring that public transport services are accessible to people with disabilities and cater to the needs of all passengers promotes inclusivity and enhances the overall user experience.

8. **Public-Private**

Partnerships (PPPs): Collaborating with private entities for the development and operation of public transport systems can bring in additional expertise and funding, facilitating the implementation of innovative solutions and improving service quality.

By integrating these components and strategies, authorities can work towards creating an efficient, sustainable, and user-friendly public transport system that meets the diverse needs of urban and suburban communities.

RFID SENSOR

incorporating machine learning algorithms to improve arrival time prediction accuracy based on historical data and traffic conditions is a valuable and practical use case for enhancing transportation and logistics systems. Here's a step-by-step approach to implementing such a system:

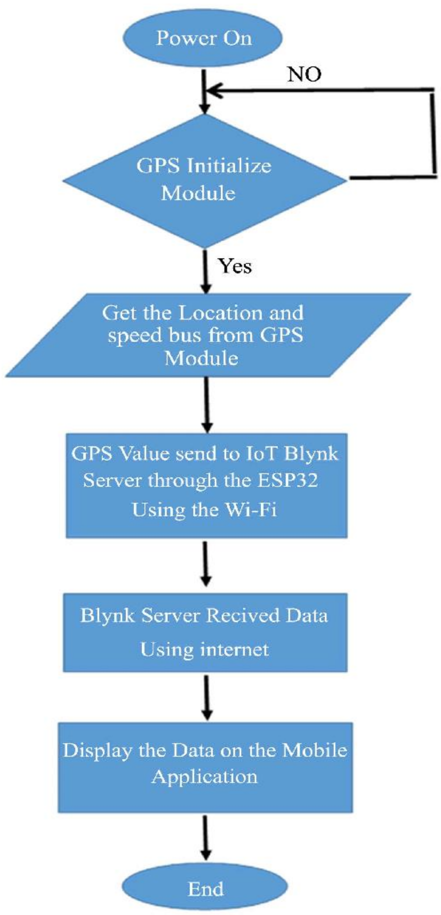
1. **Data Collection and Pre-processing:**
   - Gather historical data: Collect data on past trips, including start and end times, routes taken, traffic conditions, and any other relevant factors (weather, road closures, special events, etc.).
   - Real-time data: Integrate real-time data sources such as traffic cameras, GPS data, and weather forecasts to provide up-to-the-minute information.
   - Data pre-processing: Clean and pre-process the data, handling missing values and outliers, and converting categorical variables into numerical formats.
2. **Feature Engineering:**
   - Extract relevant features: Create features from the data that can be used for prediction, such as time of day, day of the week, road type, historical traffic congestion patterns, and more.
   - Feature selection: Use techniques like feature importance analysis or dimensionality reduction to select the most informative features.
3. **Model Selection:**
   - Choose appropriate machine learning algorithms for regression or time series forecasting. Common choices include:
     - Linear Regression
     - Random Forest Regression
     - Gradient Boosting (e.g., Boost, Light)
     - Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks for sequence data.
   - Experiment with different models and hyperparameters to find the best-performing one.
4. **Training and Validation:**
   - Split the data into training, validation, and test sets.
   - Train the machine learning model on the training data and validate its performance on the validation set.
   - Use metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) to evaluate the model's accuracy.
5. **Hyperparameter Tuning:**
   - Fine-tune the model's hyperparameters using techniques like grid search or Bayesian optimization to optimize its performance.
6. **Real-time Prediction:**
   - Deploy the trained model to make real-time predictions for upcoming trips. Integrate it with the transportation system to provide accurate arrival time estimates to users.
7. **Continuous Monitoring and Retraining:**
   - Regularly update the model with new data to adapt to changing traffic patterns and conditions.
   - Monitor the model's performance and retrain it as needed to maintain accuracy.
8. **User Feedback Integration:**
   - Collect feedback from users about the accuracy of the predictions.
   - Use this feedback to further improve the model and address any issues or discrepancies.
9. **Scalability and Deployment:**

- Ensure that the system can handle a high volume of prediction requests, especially during peak traffic times.
- Consider cloud-based solutions for scalability and reliability.

10. **Privacy and Data Security:**
    - Implement robust data security and privacy measures to protect sensitive information, such as user locations and trip histories.
11. **Regulatory Compliance:**
    - Ensure that the system complies with local regulations and data protection laws, especially when dealing with user data.

By following these steps, you can create a robust machine learning-based arrival time prediction system that continually improves its accuracy and provides valuable information to users for better trip planning and transportation management.

FLOWCHART:




1.index.html

<!DOCTYPE html>

<html>

```
<head>
    <title>Public Transport Optimization</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
    integrity="sha512-
xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAshOMAS6/ke
qq/sMZMZ19scR4PsZChSR7A=="
    crossorigin="" />
<link rel="stylesheet" href="https://unpkg.com/leaflet-control-
geocoder/dist/Control.Geocoder.css" />
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></s
cript>
</head>
<body>
    <header>
        <h1>Public Transport Optimization</h1>
    </header>
    <main>

        <h1 class="text-center">Transport  Location & Information</h1>
        <div id="map1" class="row">
        <div id="bus-info" class="col-md-3">
            <h2>Bus Information</h2>
            <p>Bus Number: <span id="bus-number">TN-2434</span></p>
            <p>Bus Name: <span id="bus-number">SETC</span></p>
            <p>Arrival Time: <span id="arrival-time">5 minutes</span></p>
            <p>Riders on Board: <span id="riders-on-board">25</span></p>
            <a href="#map" class="btn btn-primary">Location</a>
        </div>
        <div id="bus-info" class="col-md-3">
```

```
            <h2>car Information</h2>
            <p>car Name: <span id="bus-number">BMW</span></p>
            <p>car Number: <span id="bus-number">233</span></p>
            <p>Arrival Time: <span id="arrival-time">5 minutes</span></p>
            <p>Riders on Board: <span id="riders-on-board">25</span></p>
            <a href="#map" class="btn btn-primary">Location</a>
        </div>
        <div id="bus-info" class="col-md-3">
            <h2>Bike Information</h2>
            <p>Bike Name: <span id="bus-number">DUKE</span></p>
            <p>Bike Number: <span id="bus-number">TN AK -3453</span></p>
            <p>Arrival Time: <span id="arrival-time">5 minutes</span></p>
            <p>Riders on Board: <span id="riders-on-board">25</span></p>
            <a href="#map" class="btn btn-primary">Location</a>
        </div>
    </div>
        <div id="map" class="mt-5"></div>
        <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"
            integrity="sha512-
XQoYMqMTK8LvdxXYG3nZ448hOEQiglfqkJs1NOQV44cWnUrBc8PkAOcXy20w0vl
aXaVUearIOBhiXZ5V3ynxwA=="
            crossorigin=""></script>
        <script src="https://unpkg.com/leaflet-control-
geocoder/dist/Control.Geocoder.js"></script>
    </main>

    <script src="script.js"></script>
</body>
</html>
```

**2.style.css**

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
}

header {
    background-color: #007bff;
    color: white;
    text-align: center;
    padding: 20px;
}

h1{
    padding: 20px;
    margin-left: 70px;
}

main {
    max-width:100%;
    margin: 20px auto;
    background-color: white;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

#bus-info {
    border: 1px solid #ddd;
    margin: 10px 10px;
```

```
    float: left;
}

#map {
    width: 100%;
    height: 50vh;
}
#map {
    width: 100%;
    height: 50vh;}
```

**3.script.js**

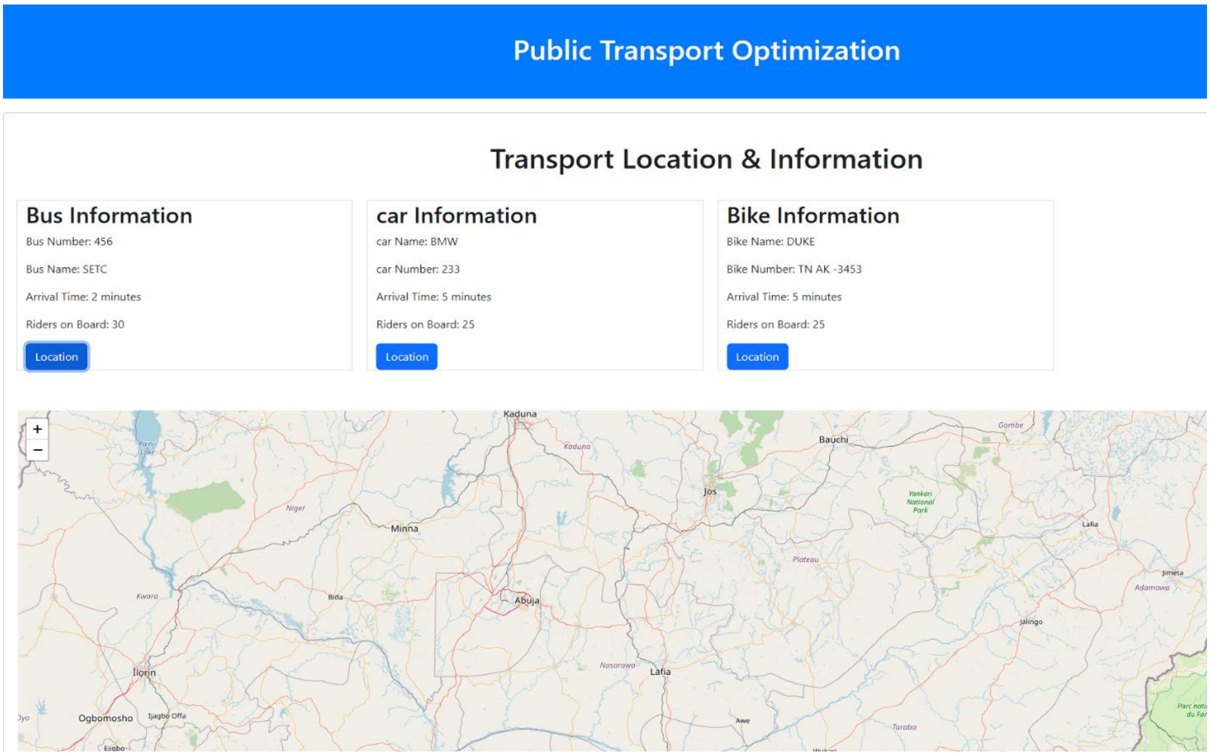
```
document.addEventListener("DOMContentLoaded", function () {
    const busNumberElement = document.getElementById("bus-number");
    const arrivalTimeElement = document.getElementById("arrival-time");
    const ridersOnBoardElement = document.getElementById("riders-on-board");

    function updateBusInfo() {
        // Simulate real-time data updates (replace with actual data from sensors or APIs)
        const busData = {
            busNumber: "456",
            arrivalTime: "2 minutes",
            ridersOnBoard: 30,
        };

        // Update the HTML content with the live data
        busNumberElement.textContent = busData.busNumber;
        arrivalTimeElement.textContent = busData.arrivalTime;
        ridersOnBoardElement.textContent = busData.ridersOnBoard;
    }
```

```
    // Simulate real-time updates every 15 seconds
    setInterval(updateBusInfo, 5000);

    // Set up Mapbox
    mapboxgl.accessToken = 'YOUR_MAPBOX_ACCESS_TOKEN'; // Replace with
your Mapbox access token
    const map = new mapboxgl.Map({
        container: 'map1',
        style: 'mapbox://styles/mapbox/streets-v11',
        center: [-73.985349, 40.748817], // Initial center coordinates (longitude, latitude)
        zoom: 12, // Initial zoom level
    });

    // Simulate bus location
    let busLocation = [-73.985349, 40.748817]; // Initial bus location

    function updateBusLocation() {
        // Simulate bus movement (replace with actual bus location data)
        busLocation = [busLocation[0] + 0.001, busLocation[1] + 0.001]; // Update bus
coordinates
        const busMarker = new
mapboxgl.Marker().setLngLat(busLocation).addTo(map);
    }

    // Simulate real-time bus location updates every 15 seconds
    setInterval(updateBusLocation, 15000);

    // Initial data update
    updateBusInfo();
    updateBusLocation();
});
```

```
const x = document.getElementById("demo");

function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.watchPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
    x.innerHTML="Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}

var map_init = L.map('map', {
    center: [9.0820, 8.6753],
    zoom: 8
});
var osm = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {

}).addTo(map_init);
L.Control.geocoder().addTo(map_init);
if (!navigator.geolocation) {
    console.log("Your browser doesn't support geolocation feature!")
} else {
    setInterval(() => {
        navigator.geolocation.getCurrentPosition(getPosition)
    }, 5000);
};
var marker, circle, lat, long, accuracy;
```

```
function getPosition(position) {
    // console.log(position)
    lat = position.coords.latitude
    long = position.coords.longitude
    accuracy = position.coords.accuracy

    if (marker) {
        map_init.removeLayer(marker)
    }

    if (circle) {
        map_init.removeLayer(circle)
    }

    marker = L.marker([lat, long])
    circle = L.circle([lat, long], { radius: accuracy })

    var featureGroup = L.featureGroup([marker, circle]).addTo(map_init)

    map_init.fitBounds(featureGroup.getBounds())

    console.log("Your coordinate is: Lat: " + lat + " Long: " + long + " Accuracy: " +
accuracy)
}
```

**OUTPUT**




Conclusion

In conclusion, Mathematical Optimization can help public transportation systems, especially in large cities, overcome their existing challenges and unlock next-level business growth.

Generally speaking, public transport is far more efficient and environment-friendly for cities than the means of personal transport. It can be difficult at first for people to stop using their cars, but as a result it will be better for everyone.