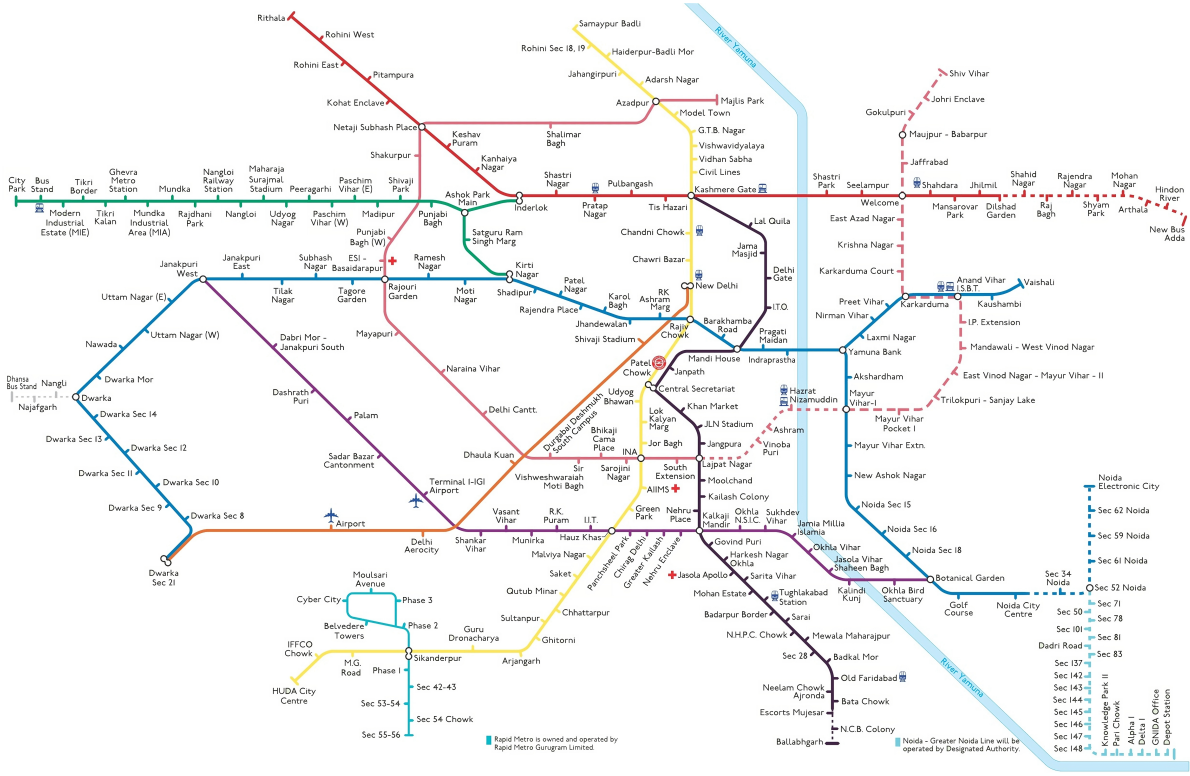
METRO API



Make a graph data structure from scratch using adjacency list.

Node → Each Station

Edge → Time & Distance b/w station

We will provide all the data, i.e. Station Names, Time taken & Distance b/w each station, Stoppage Time at every station

There are a total of *10* lines and *286* stations.

You can user **A*** or **Dijikstra** Algorithm for shortest path finding

Shortest Path Algorithm

Shortest Path refers to the minimum time it would take from source to destination station.

Note: Keep in mind, every station has a stoppage time so it will be accounted as required.

Fare Calculations

You can Calculate fare according to this table

For DMRC

Distance (in KMs)	FARE		Time Limit (in Mins.)
	Monday to Saturday	Sunday & National Holidays	
0-2	Rs 10/-	Rs 10/-	65
2-5	Rs 20/-	Rs. 10/-	
5-12	Rs 30/-	Rs 20/-	
12-21	Rs. 40/-	Rs. 30/-	100
21-32	Rs 50/-	Rs 40/-	180
More than 32	Rs 60/-	Rs 50/-	

For Rapid Metro Gurugram

Rs. 20/- only

The Train Path Finder API returns an object with the following properties:

- `linesInterchange` (array): An array of lines that are to be interchanged
    - For eg.
      From **Vaishali** to **Hindon River**
      Blue Line → Pink Line → Red Line
      `["Blue", "Pink", "Red"]`

- `interchangeStations` (array): An array of interchange stations in the shortest path.
    - For eg.
      From **Vaishali** to **Hindon River**
      Interchange would take place at **Karkarduma, Welcome**

      ```
      ["Karakarduma", "Welcome"]
      ```

      if interchange is between 1 line then array is empty, 2 lines then array has 1 value and for more than 2 lines this array would have more than 1 value.

- `startingStationName` (string): The name of the starting station.
    - For eg.
      From **Vaishali** to **Hindon River**

      ```
      ["Vaishali"]
      ```

- `path` (array): An array of station names representing the shortest path.
    - For eg.
      From **Vaishali** to **Hindon River**

      ```
      ["Vaishali", "Kaushambi", ...., "Karkarduma", "Karkarduma Court", .... , "Welcome", "Shahdara", .... "Arthala", "Hindon River"]
      ```

- `endStationName` (string): The name of the destination station.
    - For eg.
      From **Vaishali** to **Hindon River**

      ```
      ["Hindon River"]
      ```

- `timeBetweenStations` (number): The total travel time between the source and destination stations.
    - For eg.
      From **Vaishali** to **Hindon River**

      ```
      52 Minutes
      ```

- `totalFare` (number): fare charged b/w stations

- For eg.
  From **Vaishali** to **Hindon River**

  `Rs. 40`

The API calculates the shortest path based on the travel time between stations, assuming that the time between any two adjacent stations is known.