



e-ISSN: 2319-8753 | p-ISSN: 2320-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 7, July 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



Shortest Path Computation Based on Live Traffic Circumstances

A. Sreshta Reddy¹, B. Suharsha², K. Ruthwika³, K. Akhila⁴, M. Raj Kumar⁵

B.Tech. Student, Department of Computer Science and Engineering, JB Institute of Engineering and Technology,
Moinabad, Telangana, India¹

B.Tech. Student, Department of Computer Science and Engineering, JB Institute of Engineering and Technology,
Moinabad, Telangana, India²

B.Tech. Student, Department of Computer Science and Engineering, JB Institute of Engineering and Technology,
Moinabad, Telangana, India³

B.Tech. Student, Department of Computer Science and Engineering, JB Institute of Engineering and Technology,
Moinabad, Telangana, India⁴

Associate Professor, Department of Computer Science and Engineering, JB Institute of Engineering and Technology,
Moinabad, Telangana, India⁵

ABSTRACT: The online shortest path problem aims at computing the shortest path based on traffic circumstances. This is very important in modern car navigation systems as it helps drivers to make sensible decisions. There is no efficient system/solution that can offer affordable costs at both client and server sides for online shortest path computation. Unfortunately, the conventional client-server architecture scales poorly with the number of clients. A promising approach is to let the server collect live traffic information and then broadcast them over wireless network. This approach has excellent scalability with the number of clients. A new framework called traffic index is developed which enables drivers to quickly and effectively collect the live traffic information on the broadcasting channel. An impressive result is that the driver can compute/update their shortest path result by receiving only a small fraction of the index. LTI is robust to various parameters and it offers relatively short tune-in cost (at client side), fast query response time (at client side), small broadcast size (at server side), and light maintenance time (at server side) for online shortest path problem.

KEYWORDS: Broadcasting, LTI, Shortest Path.

I. INTRODUCTION

The main aim of our project is to compute the shortest path based on traffic circumstances. This is very important in modern car navigation systems as it helps drivers to make sensible decisions. There is no efficient system/solution that can offer affordable costs at both client and server sides for shortest path computation. A promising approach is that admin updates the traffic information accordingly and user views the appropriate path to reach his/her destination. Computing shortest path with traffic restriction is the current problem of the project. The path necessary for the user is displayed using maps. Then admin updates the traffic information frequently. User uses updated traffic information to reach the destination. The path required by the user is displayed usually by using maps. Google Maps API is included to display path. Dijkstra's algorithm is used to compute the shortest path. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It picks the unvisited vertex with the lowest-distance, calculates the distance through it to each unvisited neighbour, and updates the neighbour's distance if smaller. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path algorithm is widely used in network routing protocols.



II. LITERATURE SURVEY

Spectral Clustering Based On The Graph Laplacian

A connection between the Cheeger cut and the second eigenvector of the graph p -Laplacian, a nonlinear generalization of the graph Laplacian. A p -Laplacian which is slightly from the one used. Has been used for semi supervised learning .The main motivation for the use of eigenvectors of the graph p -Laplacian was the generalized isoperimetric inequality. In which relates the second eigenvalue of the graph p -Laplacian to the optimal Cheeger cut. The isoperimetric inequality becomes tight as p , so that the second eigenvalue converges to the optimal Cheeger cut value.

SHARC: Fast and Robust Unidirectional Routing

Introduce SHARC-Routing, a fast and robust approach for unidirectional routing in large networks. The central idea of SHARC (Shortcuts + Arc- Flags) is the adaptation of techniques developed for Highway Hierarchies to Arc- Flags. In general, SHARC-Routing iteratively constructs a contraction-based hierarchy during pre-processing and automatically sets arc- flags for edges removed during contraction. More precisely, arc- flags are set in such a way that a unidirectional query considers these removed component-edges only at the beginning and the end of a query. As a result, able to route very efficiently in scenarios where other techniques fail due to their bidirectional nature. It turned out that SHARC was a promising candidate for routing in time-dependent networks.

Shortest Path Tree Computation in Dynamic Graphs

The Dynamic Shortest Path (DSP) problem is to compute S from D . This problem either focuses on a single edge weight change, or for multiple edge weight changes, some of them are incorrect or are not optimized. The correct and extend a few state-of-the-art dynamic SPT algorithms to handle multiple edge weight updates. Hence prove that these algorithms are correct. Dynamic algorithms may not outperform static algorithms all the time. To evaluate the proposed dynamic algorithms, compare them with the well-known static Dijkstra's algorithm.

III. RELATED WORK

The important factors of OSP are considered (i) tune-in cost (ii) broadcast size (iii) maintenance time (iv) query response time. The tune in cost should be reduced because the power consumption is based on tune in cost. Due to short tune in cost, by using selective tuning the client receives more services like parking slot, weather related updates. The maintenance time is defined as the time taken to update the index based on the live traffic data. The response time should be very fast in few milliseconds. The broadcast size is similar to the discontinuation of receiving the latest index data. The traffic data received by the client is maintained and shortest path is computed using the uninformed search where the graph nodes are arranged in the ascending order of the distances from the source and the shortest path to the destination is discovered. Bi-directional search can be employed where the Dijkstra's algorithm is executed from forward and backward. But the response time is high and client receives more number of updates. Goal directed methods eliminate the edges that is not required to the shortest path, goal directed methods include ALT and arc Flags. In ALT, the distance between the nodes are pre computed using A^* search, Landmarks. Delling and Wagner introduced a paradigm for ALT [DALT] so that the edge weights can be updated in dynamic graphs. The maintenance cost is reduced. A tree structure is maintained for shortest path in Dynamic shortest path tree. In index transmission model, the traffic broadcast server

transmits the live index not the complete data various Road map hierarchical methods like reach , highway hierarchies (HH), contraction hierarchies (CH) , and Transit node routing (TNR) .In HH, CH, TNR the query response time is fast because the shortcuts are calculated and stored in the index. In TNR, the shortest path is computed based on the two transit nodes $A(S)$ from source and $A(t)$ from destination. The maintenance time is high because there is no efficient method to update the pre computed data. The index structure is maintained in hierarchical way in hierarchical index structures. In Hierarchical multi graph model (HiTi) the index is maintained in hierarchy not the road network. But the high maintenance time and broadcast size are the disadvantages. In Hierarchical encoded path view [HEPV], the nodes and edges are maintained in large graph which is split into smaller graphs for shortest path computation. But the index storage is a big problem.



IV. PROPOSED SYSTEM

It is a method for searching shortest path for a source and destination based on live traffic circumstances using Live Traffic Index (LTI) and Dijkstra’s algorithm.

LTI efficiently maintains index for live traffic circumstances by using Dynamic Shortest path Tree (DSPT) and hierarchical index techniques.

LTI provide short tune-in cost (at client side), fast query response time (at client side), small broadcast size (at server side) for OSP.

4.1 System Architecture

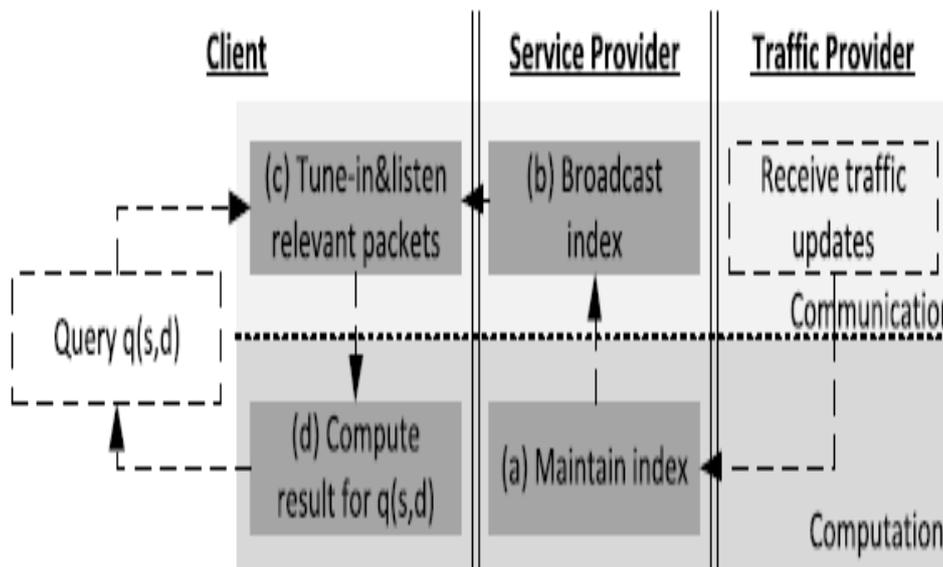


Fig. Architecture

4.2 Module Description

4.2.1 Traffic Provider

Real-time traffic feeds are collected from the traffic monitoring resources planted along the roadside and other crowd-sourcing techniques by the provider. The collected information is shared to different service provider both as raw and optimized data. This paper provides the live feed to the service providers which are collected from the advanced real-time traffic simulator. This channel provides a pre-processed traffic data to the Service provider.

4.2.2 Service Provider

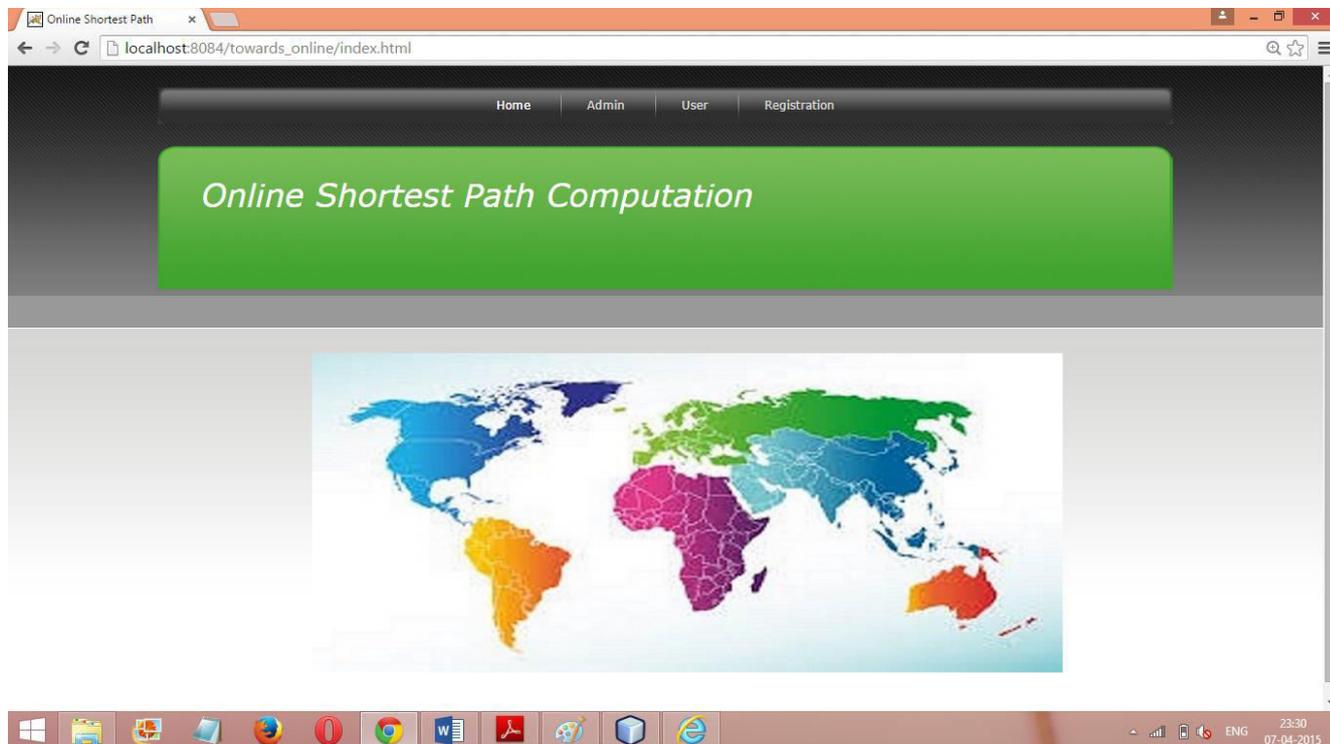
The pre-processed raw traffic data collected from the providers are being indexed and are orderly broadcasted towards multiple clients. The traffic datum are organized by indexing the collected information such as to provide real-time feeds to the clients, and also in the manner to ease the client with their computation cost.

4.2.3 Navigation Client

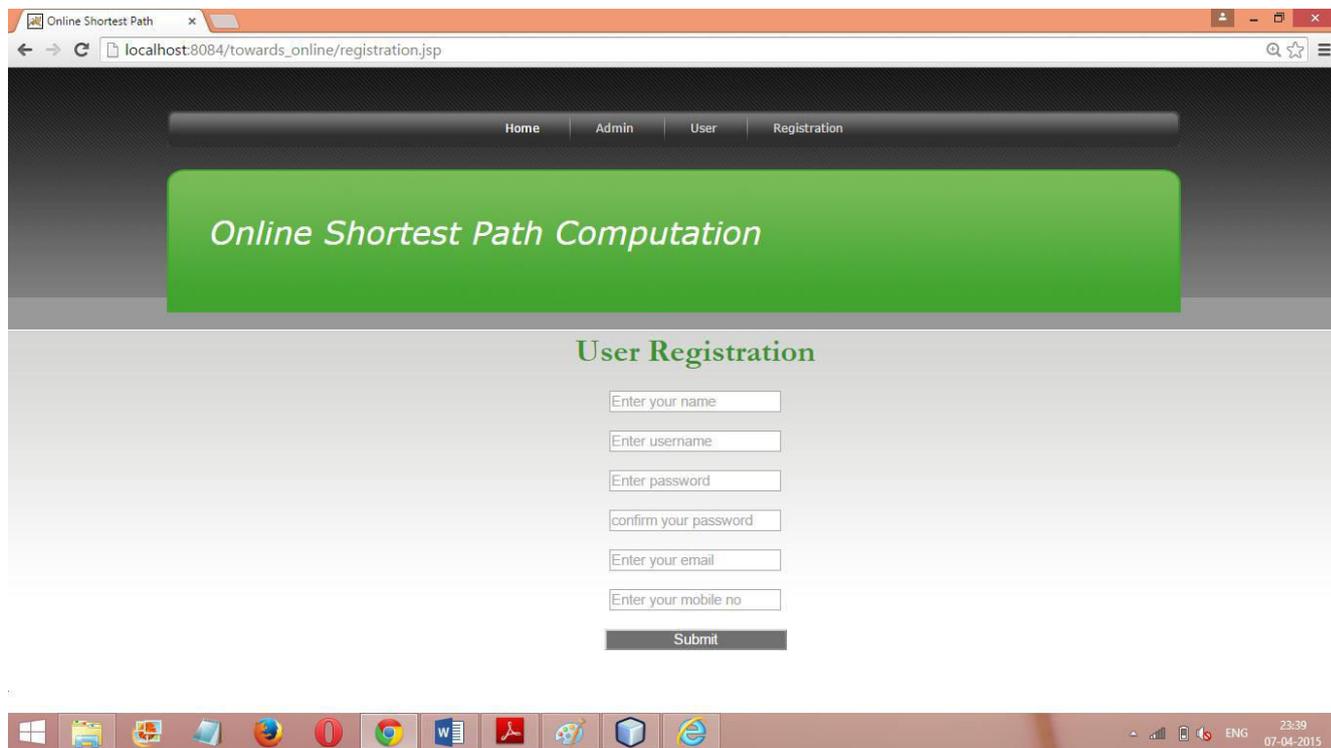
The clients which are in need of the real-time traffic feeds are tuned in a manner to receive the relevant data packets based on the indexed traffic data. The received packets are then used for computing the shortest path query, q (s, d). Generally this computation part is carried over in the server but here it does it in the client part to reduce computation overload for the server and to reduce significant computation cost.



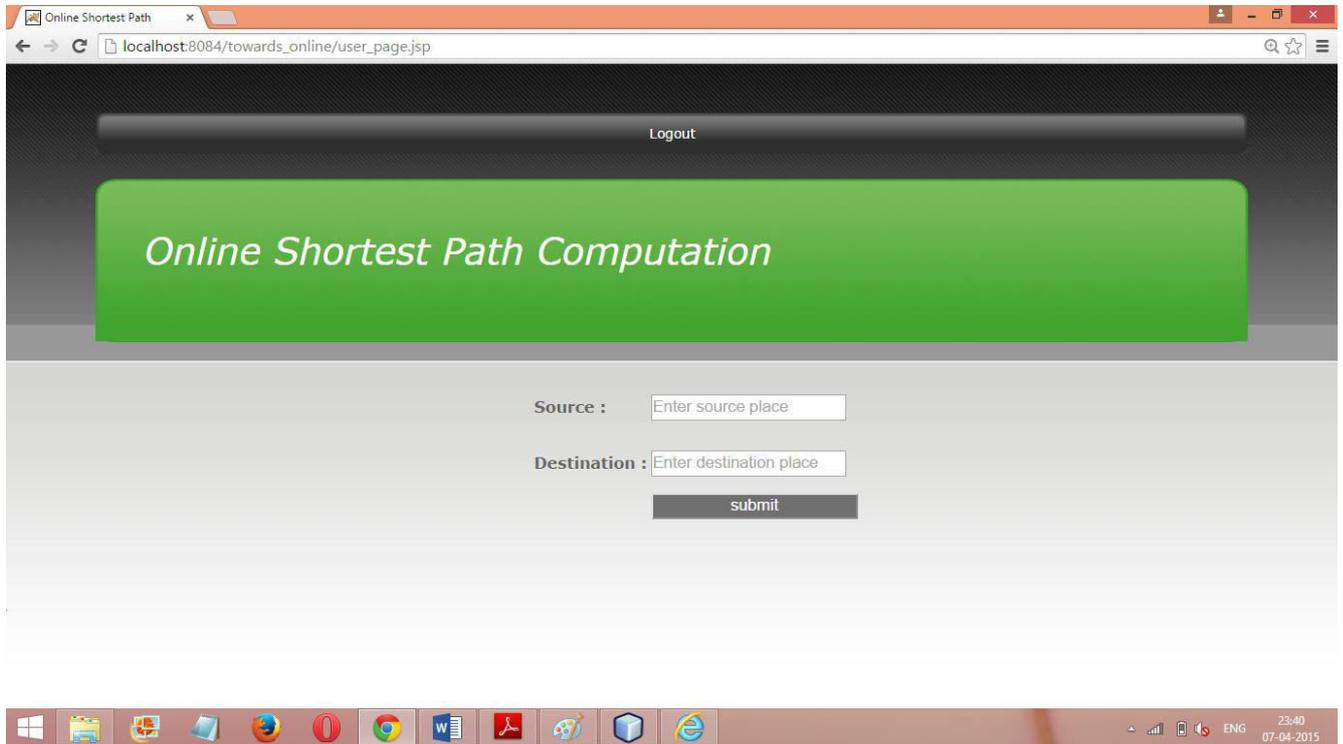
V. EXPERIMENTAL RESULTS



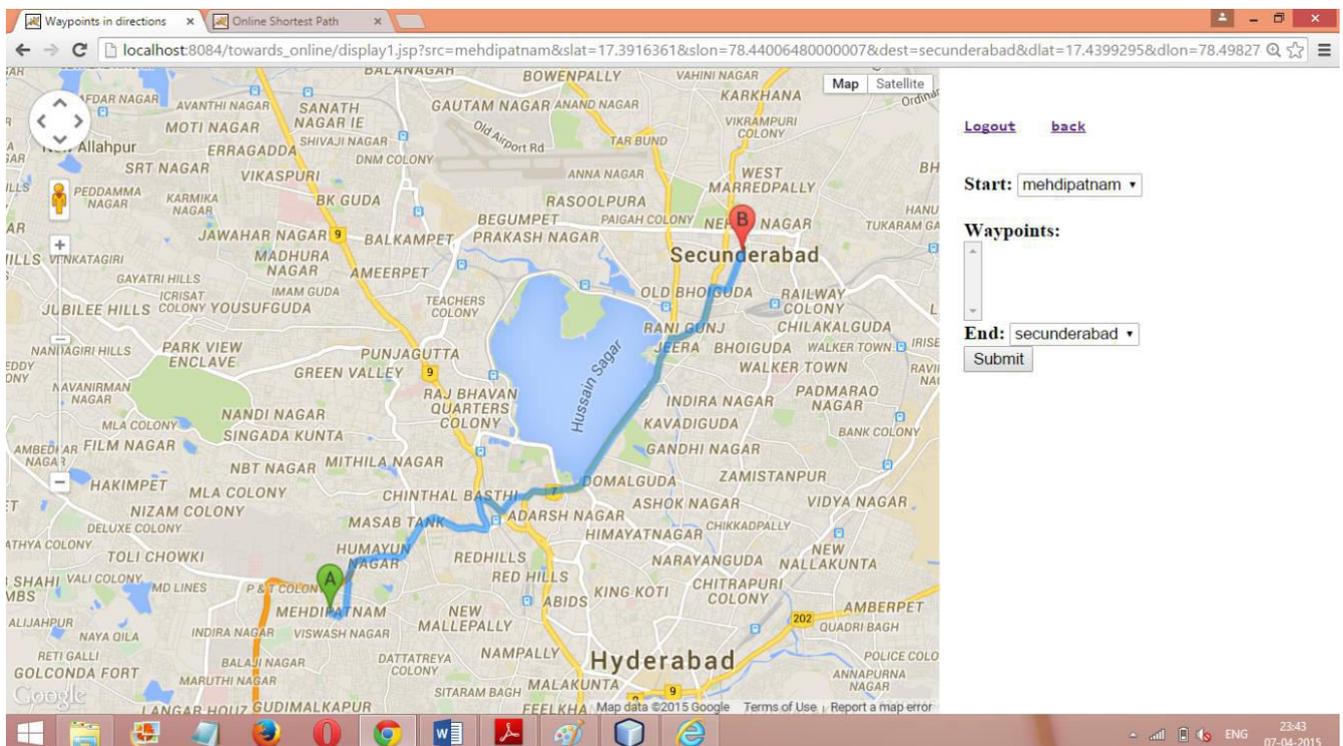
Home Page



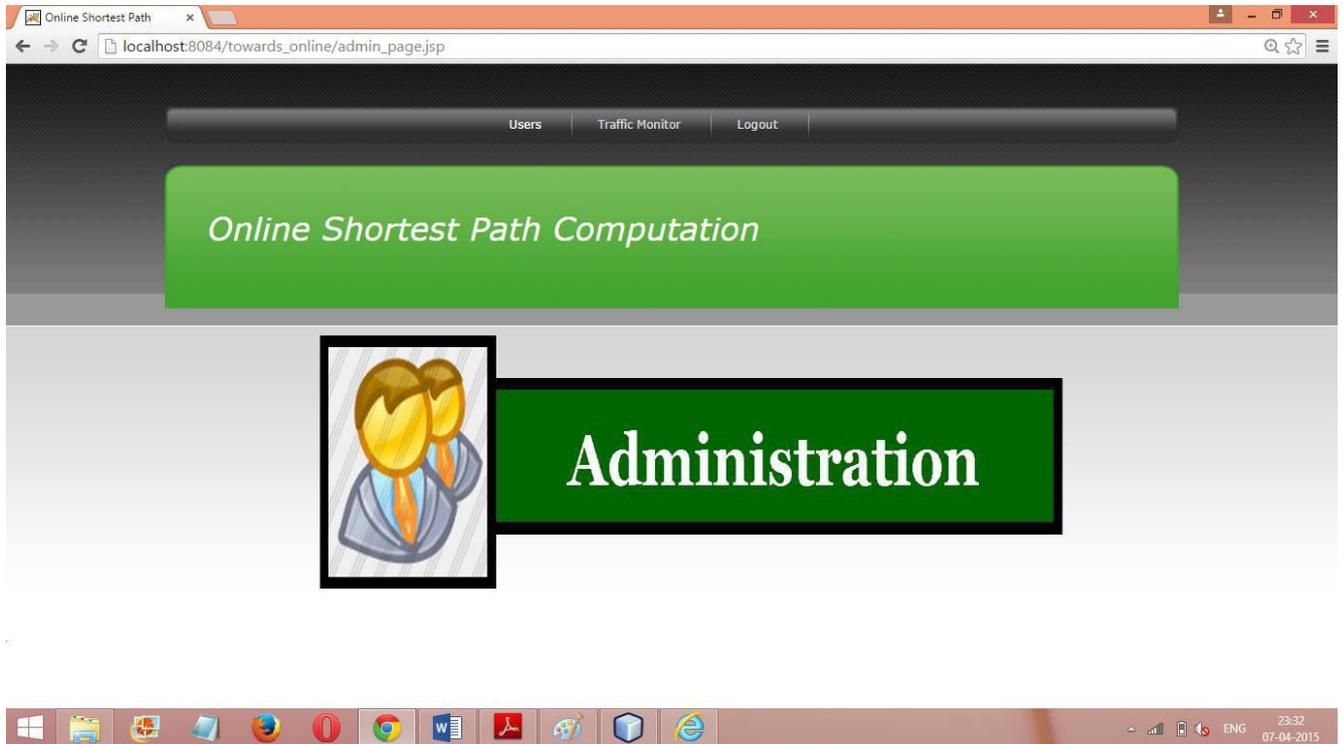
User registration form



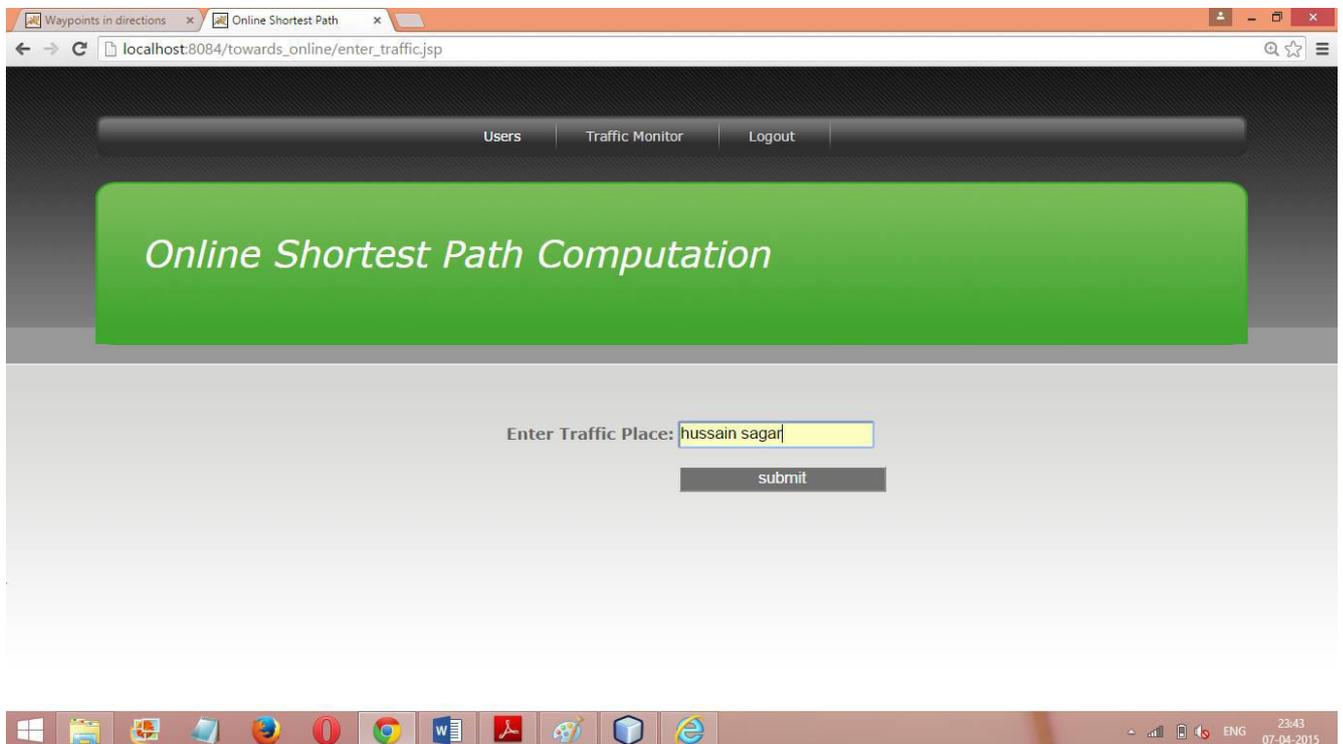
User enters source and destination to view the path



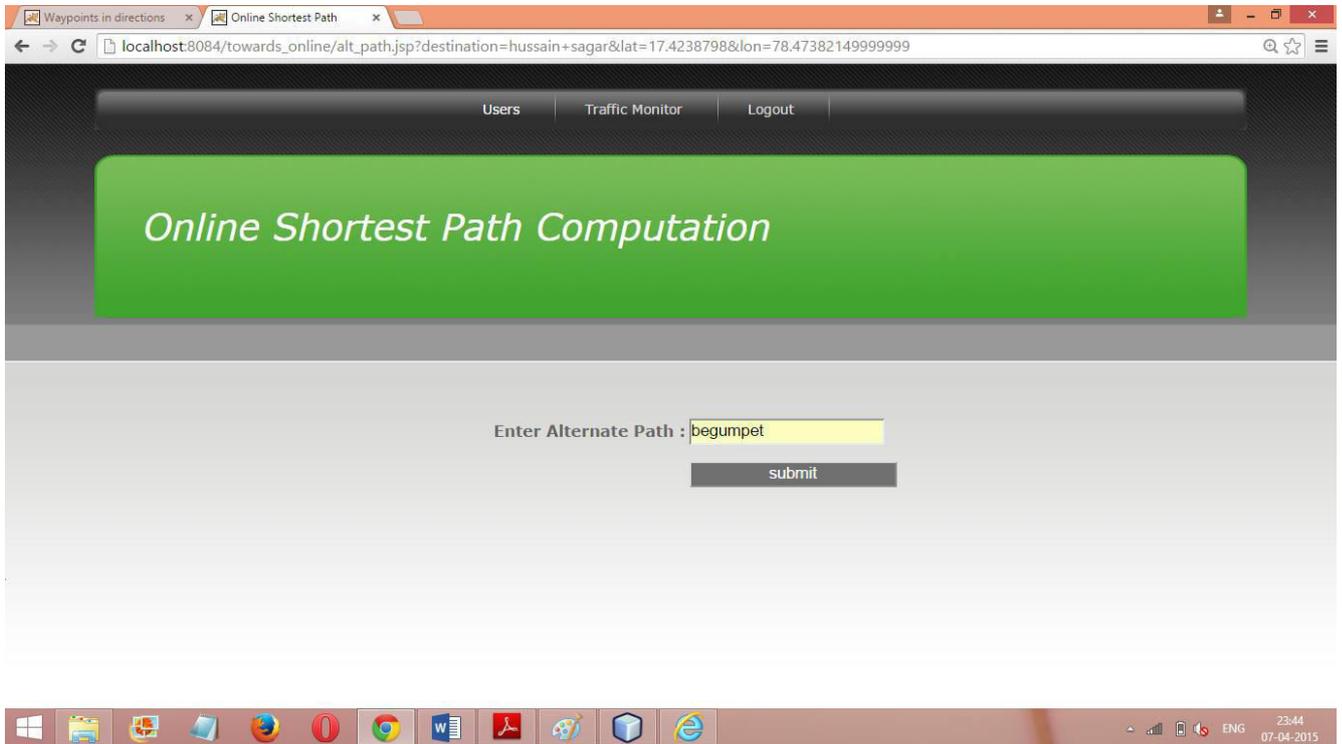
Display of the path to the user



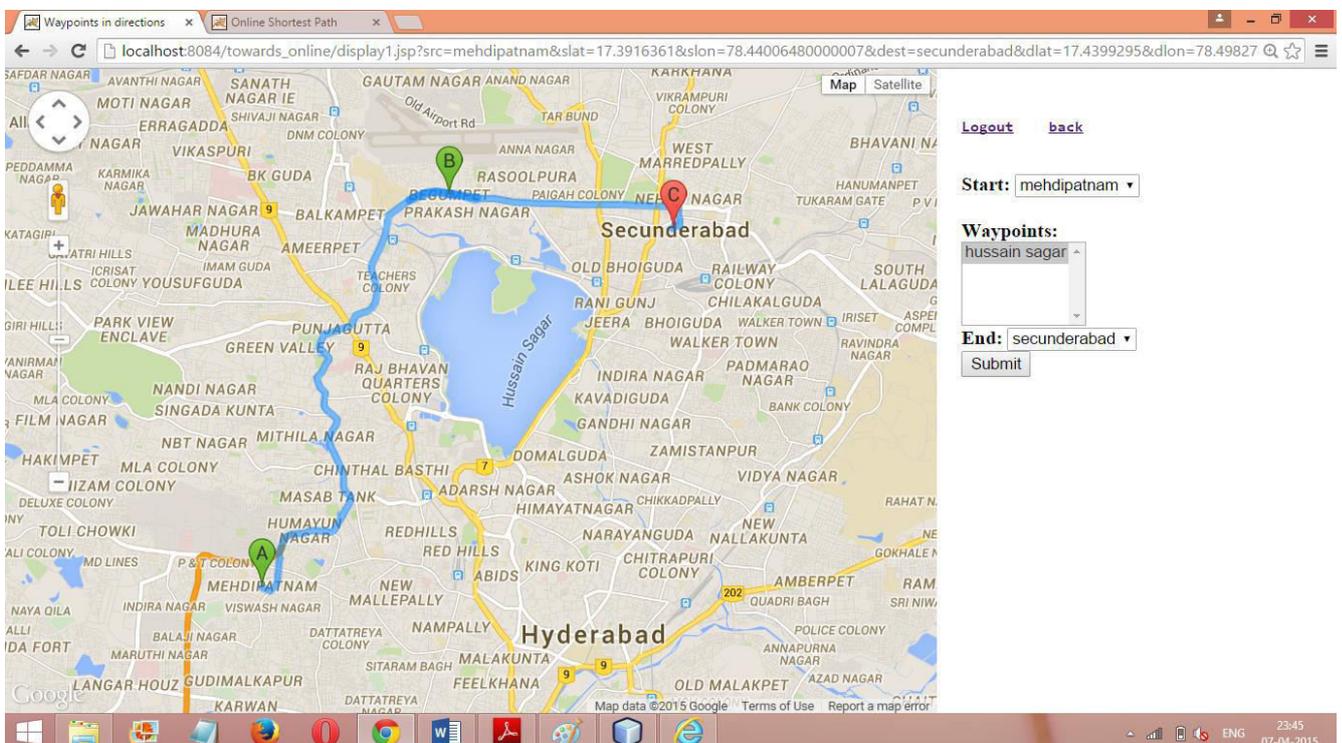
Admin views user information and enters traffic area and alternate path



Entering traffic area



Entering alternate path



Displaying shortest path to user



VI. CONCLUSION

This project is of tremendous use to many users who wish to reduce their time for searching appropriate path they are looking for. The key feature of this system is calculating the shortest path based on traffic circumstances. User initially enters source and destination to view the shortest path. Admin updates the traffic information frequently. Thus user reaches the destination as soon as possible. In this paper we studied online shortest path computation; the shortest path result is computed/updated based on the traffic circumstances. We carefully analyse the existing work and discuss their inapplicability to the problem (due to their prohibitive maintenance time and large transmission overhead).

VII. FUTURE ENHANCEMENT

In the future, we will extend our solution on effectively construct LTI and broadcast it over the air. This is a very interesting topic since the decision of a shortest path depends not only on current traffic data but also based on the predicted traffic circumstances.

REFERENCES

- [1] Leong Hou U, Hong Jun Zhao, Man Lung Yiu, Yuhong Li, and Zhiguo Gong “Towards Online Shortest Path Computation” IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 4, APRIL 2014.
- [2] www.wikipedia.com
- [3] <http://ieeexplore.ieee.org/xpl/articleDetails.jsp>
- [4] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes, “In Transit to Constant Time Shortest-Path Queries in Road Networks,”
- [5] <https://cs.uwaterloo.ca/~epfchan/publication/sptdg.pdf>- Shortest Path Trees Computation in Dynamic Graphs.
- [6] “Google Maps,” <http://maps.google.com>, 2014.
- [7] R. Goldman, N. Shiva Kumar, S. Venkata Subramaniam, and H. Garcia-Molina, “Proximity Search in Databases,”
- [8] G. Kellaris and K. Mouratidis, “Shortest Path Computation on Air Indexes,”
- [9] F. Zhan and C. Noon, “Shortest Path Algorithms: An Evaluation Using Real Road Networks,” Transportation Science
- [10] N. Malviya, S. Madden, and A. Bhattacharya, “A Continuous Query System for Dynamic Route Planning,
- [11] http://www.researchgate.net/publication/221089071_Improving_the_index_structure_with_hierarchical_techniques_in_time-series_databases



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details