

Journey Planner Implementation Essay

The screenshot shows the EssayPro website homepage. At the top, the navigation menu includes 'How To Order', 'Reviews', 'About Us', and 'Write My Essay'. The user is logged in as 'DBA: EPRO' and has options for 'Log In' and 'Sign Up'. The main banner features a student sleeping at a desk with the text: 'WRITING SERVICE AT YOUR CONVENIENCE', 'You - Send us your homework', 'We - Do it all for you', and 'Grab your original paper for just \$10 per page with a free plagiarism report included'. A 'Write My Essay!' button is present. A 'Calculate the price' calculator is overlaid on the right, showing options for 'Writing', 'Rewriting', and 'Editing', with 'Writing' selected. The calculator is set for 'Essay (any type)', 'College', '2 weeks', and '1 page / 275 words'. It offers 'Double spaces' (selected) or 'Single spaces' for a price of '\$11.4'. A 'Write My Paper' button is at the bottom of the calculator. Below the banner, there are three review sections: 'EssayPro Reviews' with a 4.9 rating, 'ResellerRatings' with a 4.9 rating, and 'Sitejabber' with a 4.8 rating.

ESSAYPRO How To Order Reviews About Us Write My Essay DBA: EPRO Log In Sign Up

WRITING SERVICE AT YOUR CONVENIENCE

You - Send us your homework
We - Do it all for you

Grab your original paper for just \$10 per page with a free plagiarism report included

Write My Essay!

Calculate the price

Writing Rewriting Editing

Essay (any type)

College 2 weeks

1 page / 275 words

Double spaces Single spaces

\$11.4

Write My Paper

NO MORE SLEEPLESS NIGHTS...
100% PLAGIARISM-FREE ESSAYS. ANY TOPIC OR DIFFICULTY CAN BE HANDLED!

EssayPro Reviews 4.9

ResellerRatings 4.9

Sitejabber 4.8

LINK => <http://787787.com/writing-service?311294329>

Journey Planner Implementation

This document envisages the creation of a computer software application which enables a user to identify the shortest route between two given stations on the London Underground system. The application itself, is to be written in the Ruby scripting language and the schema of the Underground is to be stored on a MySQL database. In order to be effective, the application would provide an implementation of Dijkstra's shortest path algorithm . An initial prototype of the application would be a console based application i.e. a non-gui style application. This could be further developed into a graphical user interface (gui) application or possibly even a web-based application. The application should also provide administrative options to backup and restore the database, and make use of MySQL scripts to create and populate the database schema.

Rationale behind software choices

Ruby was selected as the main scripting language because the problem space maps well to real life objects and Ruby is highly object oriented. A solution to the shortest path algorithm will spend its time scanning 'Nodes' in our case Tube line stations, and

'Edges' a connection between two adjacent stations. Ruby should make easy work of applying Dijkstra's algorithm to generate code. MySQL was selected as the target database because MySQL is open source, readily available and works well in tandem with Ruby.

Environment and Project Design

The project would be created on the Linux distribution Ubuntu and should make use of modern design practises such as entity relationship diagram modelling for the database schema. The application itself should be object oriented and be designed with the aid of The Unified Modelling La...

... middle of paper ...

... of adjacent nodes (stations) is connected both ways. Does the application still work correctly? Now include more stations/lines and test again.

Next, decide on a database design which can load the nodes and edges. Modify the Ruby code to dynamically populate the Nodes and Edge objects from the database. Once again, test thoroughly. At this stage, the core 'engine' of the application is now in place. Modify the code so that it becomes parameterised and will accept source and target station names generating suitable journeys. Develop MySQL scripts to insert all lines and stations of the [London Underground](#) system. How well does the application scale? Finally, wrap up the application as a menu driven program which prompts the users for pairs of stations against which to calculate the shortest path and provide administration options to backup and restore the database.

Other Arcticles:

- [King'S College Dissertation Binding](#)
- [How To Write Korean Characters On Facebook](#)
- [Comparing The Two Versions Of To Build A Fire](#)
- [Developing Critical Thinking Skills In Virtual Collaborative Learning Programs](#)
- [Online Homework Planners](#)
- [Custom Writing Lessons](#)
- [Layout For An Argumentative Essay](#)
- [How To Write Funny Christmas Letters](#)