Programming Task 1: For the position of a simulation programmer.

This task is for the following job description:
http://fieldsofview.in/projects/work-with-us/simulation-programmer/

From the World Bank dataset (http://data.worldbank.org/indicator), download the datasets for the following indicators

1. Education: Government expenditure on education as % of GDP (%)
2. Public Sector: Cash surplus/deficit (% of GDP) and Tax revenue (% of GDP)
3. Health: Health expenditure, total (% of GDP)

Your tasks will include the following:

1. Write a script to parse the dataset, and store it in a relational database. The relational database must be normalized to at least 2 NF. Use any appropriate DBMS. Along with the parsing script, you should mail the ER diagram for your schema.
2. Use k-means or hierarchical agglomerative clustering, to cluster the countries along the four datasets: education, cash surplus, tax revenue, and health expenditure. Please give a brief explanation of why either of the techniques was chosen. You may use any clustering tool such as cluto (http://glaros.dtc.umn.edu/gkhome/views/cluto) to perform the clustering.
    a. The variables to be used for clustering are the values from each of these datasets for each of the countries. First, classify each country, on each variable, according to %age of GDP. For example, the bins of classification could be 0-3%, 3.1-5%, 5.1-6%, 6.1% and above. Do this for each of the variables: education, cash surplus, tax revenue, and health expenditure.
    b. Now, use the different classifications for each country to represent the country as a vector. For example, for a country X, $V_X = S_e + S_t + S_c + S_h$ where V_x is the vector for the country X, S_e is the spending on education, S_t is the tax revenue, S_c is the cash surplus, and S_h is the spending on healthcare.
    c. Write a separate object oriented program which reads from the database, and performs the necessary clustering based on these variables. The results of the clustering should be stored on the same relational database, with appropriate normalisation. You should mail us the script to perform this clustering.
3. Based on the outputs of the clustering, write an interactive web-based program that produces the following charts: (You may use any object oriented framework such as Django, Rails, Symfony, etc.)
    a. Bar chart for the classifications of each of the 4 variables
    b. Scatter plot of the clusters

You may use any of the following languages: Python, Perl, C, C++, Java, C#. You may use a combination of these languages. You have to use git as the version control system, with commits at regular intervals, with appropriate comments. You have to write unit test cases wherever appropriate.

You will submit:

1. The ER diagram.
2. The SQL file with all data, including results of clustering.
3. Program to fetch data.
4. Program to cluster data.
5. Web program to produce interactive charts.

You will be evaluated on:

1. The ER diagram (20%)
2. Script 1: Fetching data, parsing it, and entering it to a RDBMS. (20%)
3. Script 2: Clustering program (Object Oriented). (20%)
4. Script 3: Script to produce interactive charts (Object oriented). (20%)
5. Standard development lifecycle (version control, documentation, etc.) (10%)
6. Unit tests. (10%)

The general rules for the exercise are below:

1. You have until **10.00 AM *7th July 2016*** to submit the exercise. Mail us the github link for your submission. Your repository must contain all the necessary parts of the submission.
2. Add a README file in the root directory of your project. This should describe the directory structure, build instructions, and the list of dependencies. **Your project must build under a GNU/Linux environment.**
3. If your project does not build using the dependencies you have mentioned, you will not be eligible for further rounds of the interview.
4. Plagiarism of any sort is strongly discouraged at FoV. We will disqualify any entries found to have plagiarized code.

Please share the **Github** URL project for your project in the email. Make sure that the last commit is before **10.00 AM *7th July 2016***. Please share the GitHub URL to work@fieldsofview.in. In the subject of the email, please add “[Application for post of Simulation Programmer]” followed by your name and affiliation. Submissions not following this format will not be evaluated.

Programming Task 2: For the position of the Game Programmer
This task is for the following job description:
http://fieldsofview.in/projects/work-with-us/game-programmer/

Using OpenGL, PyGame, or Processing, implement a single player, top-view game where the you play a cyclist, riding on the newly added cycle lanes of a city.

You will implement a small intersection with two roads, of two lanes, with minimal traffic conditions, where the vehicles follow perfect lane behaviour. The left 20% of each lane is demarked as the cycle lane. You are free to base this on any existing cycle lane implementation that you are aware of.

1. Assume that the cyclist knows how to ride a bicycle in the game. The controls should only be for the direction of navigation.
2. The game will have three levels:
    a. First level will be an elementary level, where the cyclist uses the arrow keys to navigate through a straight road.
    b. Second level will involve obstacles in the cycle lane, such as parked cars (for example).
    c. Third level will involve obstacles in the cycle lane, as well as intersections.
3. Scoring: You will need to design a scoring system for the game
    a. The score shall take into account aspects such as riding within the cycle lane versus outside it, obstacles avoided/ hit, etc.
    b. You are free to develop a scoring system which takes other aspects into consideration. Please explain your scoring system as a visual in the game.
4. The game should allow for restarting once the game is over.
5. The game should save scores, and display high scores at the end of the game. The high scores should also be available as a menu option at the beginning of the game.

You will submit:
1. The level design document.
2. The source code and executable of the game.
3. Scoring design document.

You will be evaluated on:
1. Structure of the code. The criteria will vary depending on whether you use OpenGL, PyGame, or Processing. You are not allowed to use 3D game engines for this exercise. (40%)
2. Level generation and transition (25%)
3. Scoring system which you design for the game. (25%)
4. Standard development lifecycle (version control, documentation, etc.) (10%)

The general rules for the exercise are below:

1. You have until **10.00 AM *7th July 2016*** to submit the exercise. Mail us the github link for your submission. Your repository must contain all the necessary parts of the submission.
2. Add a README file in the root directory of your project. This should describe the directory structure, build instructions, and the list of dependencies. **Your project must build under a GNU/Linux environment.**
3. If your project does not build using the dependencies you have mentioned, you will not be eligible for further rounds of the interview.
4. Plagiarism of any sort is strongly discouraged at FoV. We will disqualify any entries found to have plagiarized code.

Please share the **Github** URL project for your project in the email. Make sure that the last commit is before **10.00 AM *7th July 2016***. Please share the GitHub URL to work@fieldsofview.in. In the subject of the email, please add “[Application for post of Simulation Programmer]” followed by your name and affiliation. Submissions not following this format will not be evaluated.