



Softablitz Problem Statements

B.Tech 2nd , 3rd year And MCA 1st , 2nd year

Basic Instructions

1. A team can consist of at most 3 members.
2. All branches are allowed.
3. Plagiarism in any form will result in elimination.
4. Teams can attempt at most 2 problems.
5. Each problem is of certain points (Points will be added soon). Full points will be awarded only when all the features described in the problem statement are implemented.
6. Each problem has a specific theme, bonus points will be awarded if the final solution is in accordance with theme.
7. Bonus points will also be awarded if team implements features other than in the problem set.

Few Advices

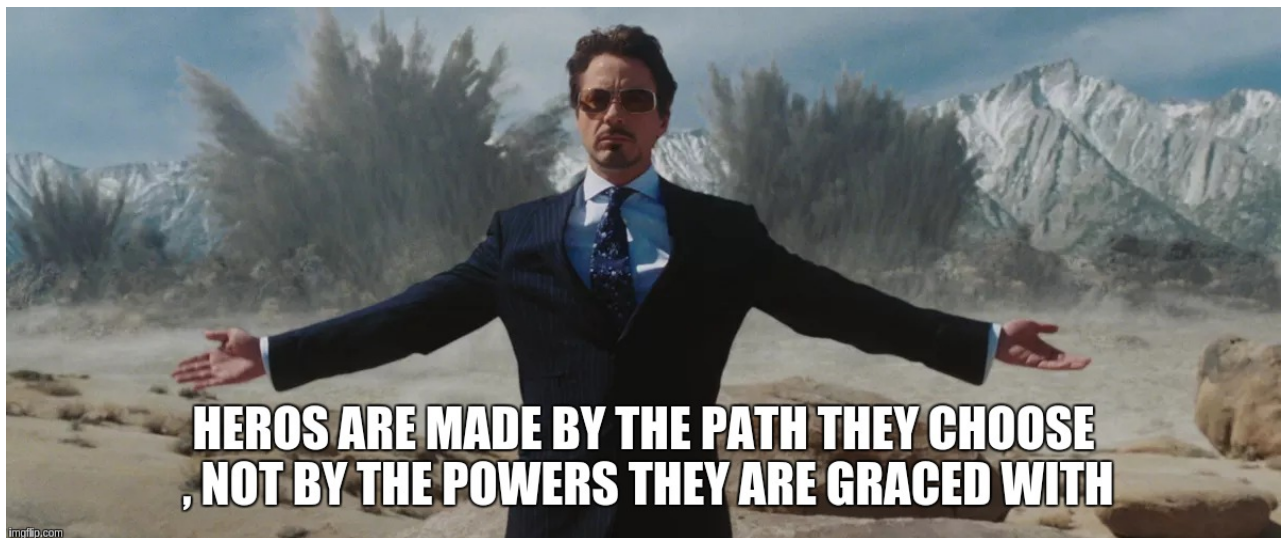
1. Firstly read each and every problem statement and then decide which one suits you best with your current skill set. Don't just run after points.
2. Try to build a complete solution for the problem that you are attempting. In this way you will have a project worth mentioning on your resume.
3. Don't try to solve 2 problems if you are sure that you won't be able to complete them fully by the end of Avishkar.
4. Although UI is very important part of any desktop application, but first try to implement as many features as you can. In case you get time time after this, then only spend time polishing UI. Remember an application with very good UI but minimal functionality is useless. It is just like, "**Naam bade par darshan chote**"
5. Last but not the least for any doubt regarding problem statements contact coordinators we are always here to help.

StarkHub

Description:

Tony Stark, our beloved Avenger is worried about content restrictions being imposed by YouTube. Being a futurist he senses a threat, that one day YouTube might have full control over all the video content. Inspired from BitTorrent, he thought of a Peer to Peer streaming service with features similar to YouTube. Tony is too busy while making his new bleeding edge armor. So he needs your help to create this platform while he prepares for a bigger threat.

PS. The team with best implementation of this software will get Stark Internship ;)



Basic Features:

1. A user account where each user can be content creator or viewer (or both).
2. Content creators can have multiple channels that they can manage.
3. Content creators can add video to any of their channels.
4. Viewers can subscribe to any channel and they must also get notified every time a new video is added to the channel they are subscribed to.
5. Content creators must keep their instance of StarkHub up and always running, so that in case a user requests a video then it must stream it. If the instance of StarkHub is down then all the videos of the respective content creator must be made temporarily unavailable.

6. Viewers can search all the videos available over the network by the video title (Almost similar to youtube search).
7. Viewer can also like, comment and add a video to their watch-later list.
8. A user profile dashboard.
 - a. If user is a content creator then he must get stats of all the videos added (Like number of likes, view count etc.) in his dashboard.
 - b. If user is a viewer then he must get log of all his activities. (More the statistics you show, more the points you get)

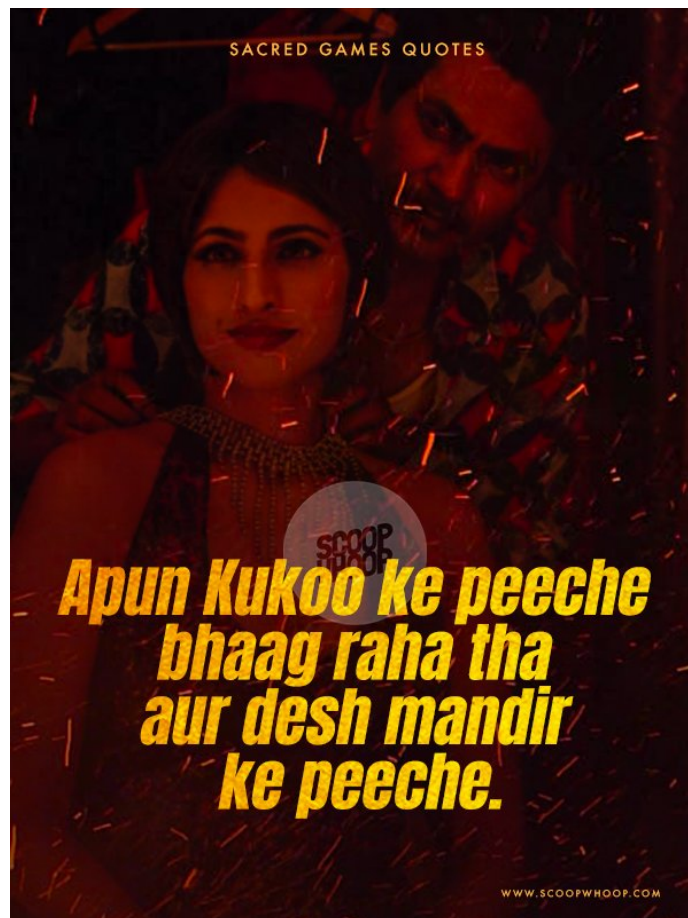
Advanced Features:

1. Content creators can also add tags to video whenever they add video.
2. Viewers can also search videos based on those tags.
3. A trending-videos page, where viewers can see the list of trending videos (For eg. You may set the criteria for trending as the number of unique views a video gets per hour)
4. Two plans for content creators - free and premium.
 - a. Free plan - content creator have to keep their instance of StarkHub up and running(if their instance goes down, their content will be immediately unavailable) .
 - b. Premium plan - a copy of all the content added by the creator(with premium membership) will be stored at one extra peer (who may be a content creator or a content viewer) and in case of downtime of the original creator's instance, this cache-peer would be used for streaming purposes. Ensure that the cache-peer on which the data is stored as backup cannot modify it. (You can use a checksum for achieving this)
5. Provide an overall rating for every channel based on the overall downtime, number of subscribers and several other factors.
6. Recommend videos to viewers based on the videos they previously watched and the channels they have subscribed. (You are free to explore any recommender system API for this feature)
7. Bonus points for UI that follow Material Design Guidelines.

Sacred Bazaar

Description:

“Apun Kukoo ke peeche bhag raha tha aur desh mandir ke peeche!!”. Gaitonde’s love for Kukoo was eternal. He loved her/him conditionally. Therefore as a last gift to her/him, Gaitonde aimed to buy her/him a lot of products which she/he liked. Help Gaitonde to use the perfect Sacred Bazaar portal.



Basic features

User features

1. A user can sign up for the retailer.
2. A user should be able to login and logout of the system.
3. He can shop for a certain session in which he is logged in and place the order.

4. A user should be able to view his shopping history for a given day and given month.
5. He can query information about any product in the inventory.
6. A user should be able to view the total amount spent up to date in the various transactions he has been involved with.
7. User can query the offers related to a particular product.
8. User should be able to get a sorted list of products based on prices.

Admin features

1. Maintain customer information (whatever you can think of :P)
2. Maintain product information like MRP, manufacturing date, expiry, type (like food, washing product, grooming, clothes etc.), date at which it was bought etc.
3. Retrieve the various time moments when a product of certain type was sold.
4. Generate the bill for a certain order placed by the client and send it to client.
5. Multiple clients can place the order at the same time.
6. Apply offers (x % off, buy X get Y free, etc) on certain type products over a time duration.
7. Maintain the quantity of products and increment/decrement them as per need.
8. Add/remove a particular type of product.

Advanced features

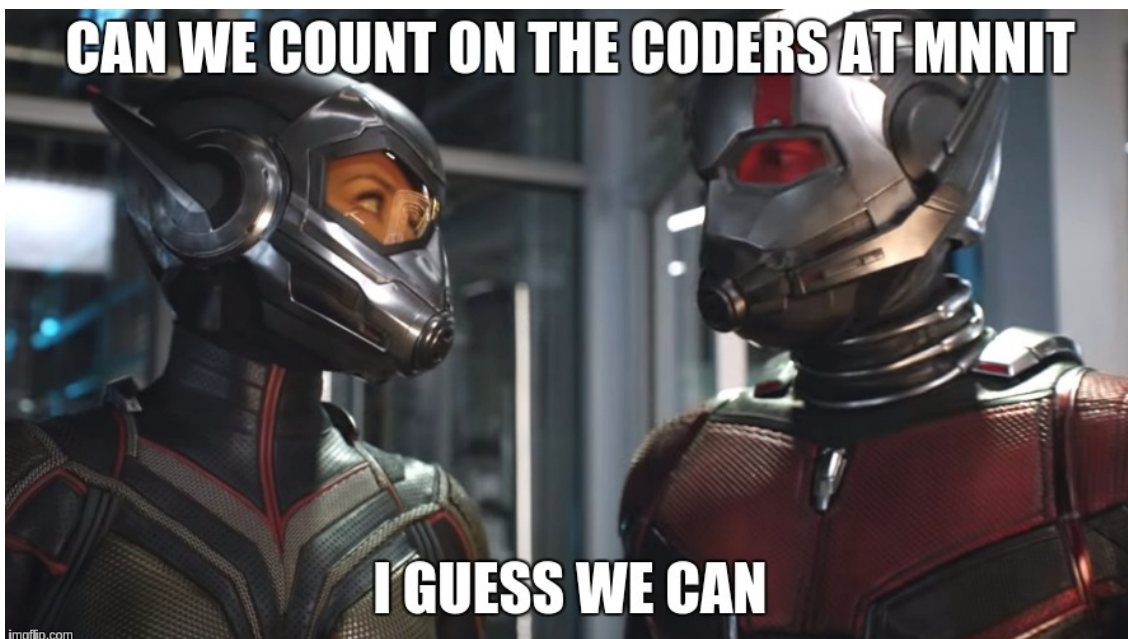
1. Discounts always attract users. Give a discount to those users who have bought worth of certain X rupees for the current month. This discount will apply for certain number of months that can be set by the admin.
2. For the discount scheme, users should get notified about the scheme.
3. A retailing service would always want to analyze itself. Therefore prepare charts/graphs to depict the rate of a particular type of product sold over a certain time period.
4. Identify the top 5 customers which have the highest transaction volume for a certain month and give them a heavy discount percentage for a certain time duration.
5. Prepare graphs/charts to analyze a user's interaction with the retailer over a period of time.
6. Get any given user's transaction history over any period of time.

Squiz (Secure Quiz)

Description:

Professor Hank Pym takes a weekly quiz to check his students' progress, but unfortunately he is always so busy in helping AntMan and the Wasp in their adventures that he hardly gets any time to assess them. So he thought of making 'Squiz' - a secure quizzing platform which will help him, not just in organizing quizzes but also in assigning grades to them. Professor Pym wants you to create Squiz, as he is busy creating the Quantum Tunneling device to bring his wife back.

PS. Professor Pym has promised to give Quantum Healing Particle to the team with best implementation.



Basic Features:

1. Create a Teacher/Student Account.
2. Teacher can add new subject and host multiple quizzes under one subject.
3. Teacher can add various sections to a quiz and allot different time limit to each section.
4. Teacher can also add solution to quiz.
5. Multiple students can login at the same time (From their own devices) and give the quiz.

6. All the students should have the same set of questions in one section but in random order.
7. There are individual timers for each section and the students can switch between the different sections throughout the exam.
8. The student is shown with the total points after the completion of the exam, if the teacher has added the solution by the time student submitted his solution otherwise show the status as pending for evaluation.
9. The students can ask a query to the teacher to which the teacher may/may not respond.

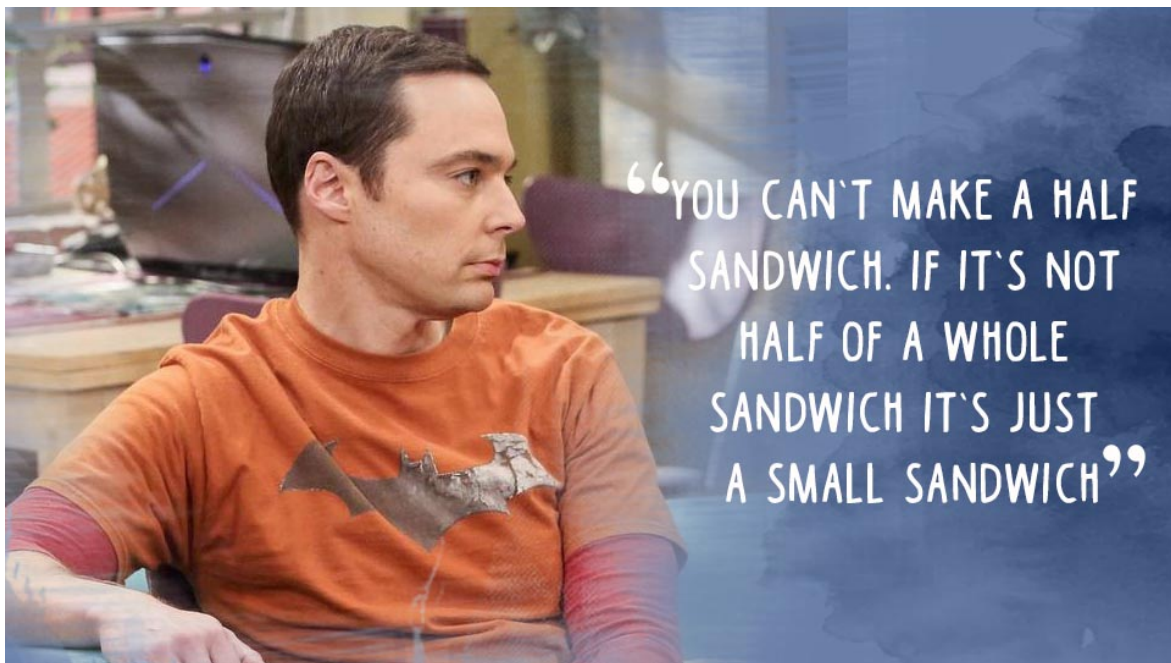
Advance Features:

1. The questions can be of different types(Single Choice Correct MCQs, Multiple Choice Correct MCQs, T/F).
2. The students can give a rating to the quiz and the platform.
3. A profile dashboard for both students and teacher where they can see all the final ranking (Once quiz has been graded) and other statistics like quiz overall rating.(More the statistics, more the points)
4. All the network operations like transfer of quiz questions and responses must be encrypted.
5. In case of a connection breakdown all the answers should be stored in a file/local cache (Again in encrypted form) and restored automatically when the connection is re-established. (Prefer storing in local in memory cache like redis).

The Wolowitz terminal

Description:

Sheldon Cooper is the geeky science nerd we all are acquainted with. He has his way of annoying his flatmates all the time, especially the sweet Penny. The prime source of annoyance is his geeky talk all day long. Now he plans to expand his knowledge to that of a Computer Science Maestro. For that he must build a sexy terminal for his personal use, as all the computer science masters do. He names the terminal after his friend Howard - WoloWitz TTY. Help Sheldon to build the most beautiful, fascinating and powerful terminal ever, that does justice to his reputation - The king of Geeks.



Note: Don't use the 'system()' function call to execute commands. That'll not be counted. Also do not make processes using ProcessBuilder, that again will not be counted. Implement functionality of commands using your own logic and using the Java system library. The same applies for users that are using any other language for development.

Basic features:

1. A user will first login to the terminal.
2. Implement the basic file management commands. Examples: ls, pwd, mkdir, cd etc. Explore the file API that is associated with the standard library.

3. Maintain information about the current user that is logged in. Examples: `user add` etc.
4. Implement basic system information commands, like displaying OS name, displaying IP address etc. Explore the library that is associated with these functionality.
5. Try to implement hardware information.

Advanced features:

1. Maintain information about the various groups that a user is involved with.
2. Try to implement the concept of permissions on files and directories. Only authenticated users can open those files and directories.
3. Implement functionality to open compressed files.
4. Implement text processing/searching commands. Use regular expressions for the same.
5. Implement your own commands that can be of good aid.
6. Bonus points for UI, themes etc.

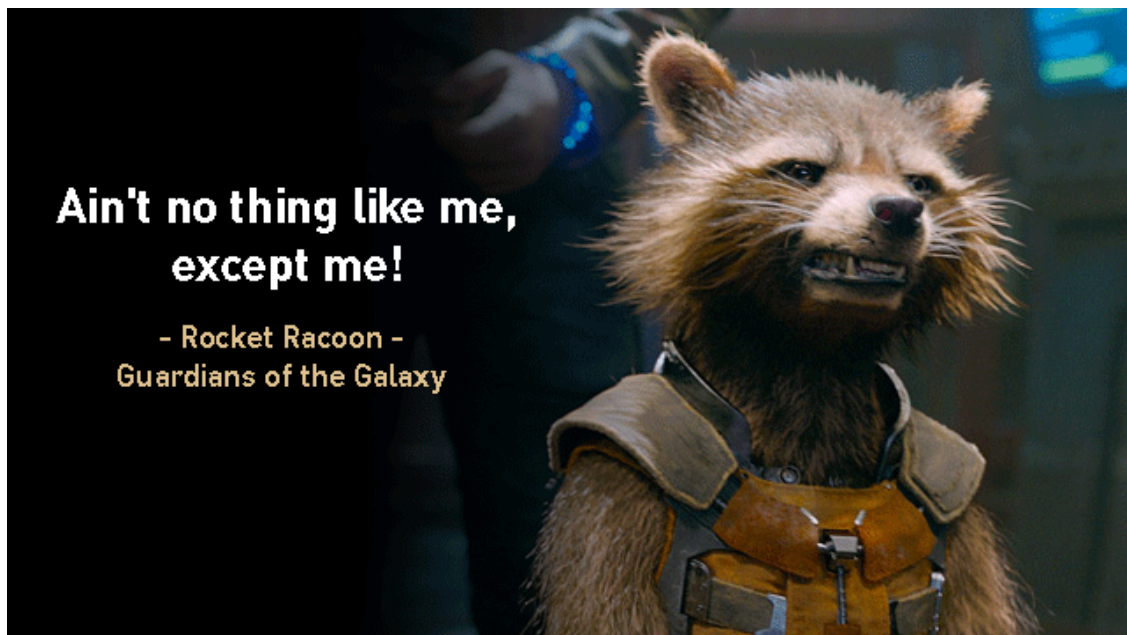
Note: You are not required to implement all the commands that may cover the above mentioned features. Implement as much as you can and some good implementations will render appreciation. :)

Awesome Player

Description:

Star Lord loves to hear Awesome Mix(Volume 1) on his Sony Walkman but unfortunately it broke during an intergalactic fight. Since Star lord can't fight without listening to music, Rocket decides to help his friend. He had earlier stolen a Windows tablet for Star Lord from The Collector's collection. But unfortunately, this tablet didn't have any music player installed on it. Rocket can't risk wasting time to search a new device for Star Lord, as he needs him back on the field ASAP. So he has asked the great coders of MNNIT to build a Music Player for that tablet. (There is no internet in space. The rabbit can't code so you Gotta code everything yourself) He will be arriving at MNNIT on August 27th. So be prepared with the music player.

PS. Rocket has also promised Guardian Of Galaxy membership to one team that implements the best Music Player.



Basic Features:

1. Should extract music files from all over PC. (not only MP3)

2. Search by the name of the song, artist, genre & album.
3. Play, pause, next, previous with seekbar.
4. Should contain the list of songs most played & recent played.
5. Should be able to shuffle & sort according to name & artist.
6. Categorisation based on artists, genre & album.
7. Should be able to create playlist. (Without copying into separate folders)
8. Volume adjuster.

Advanced Features:

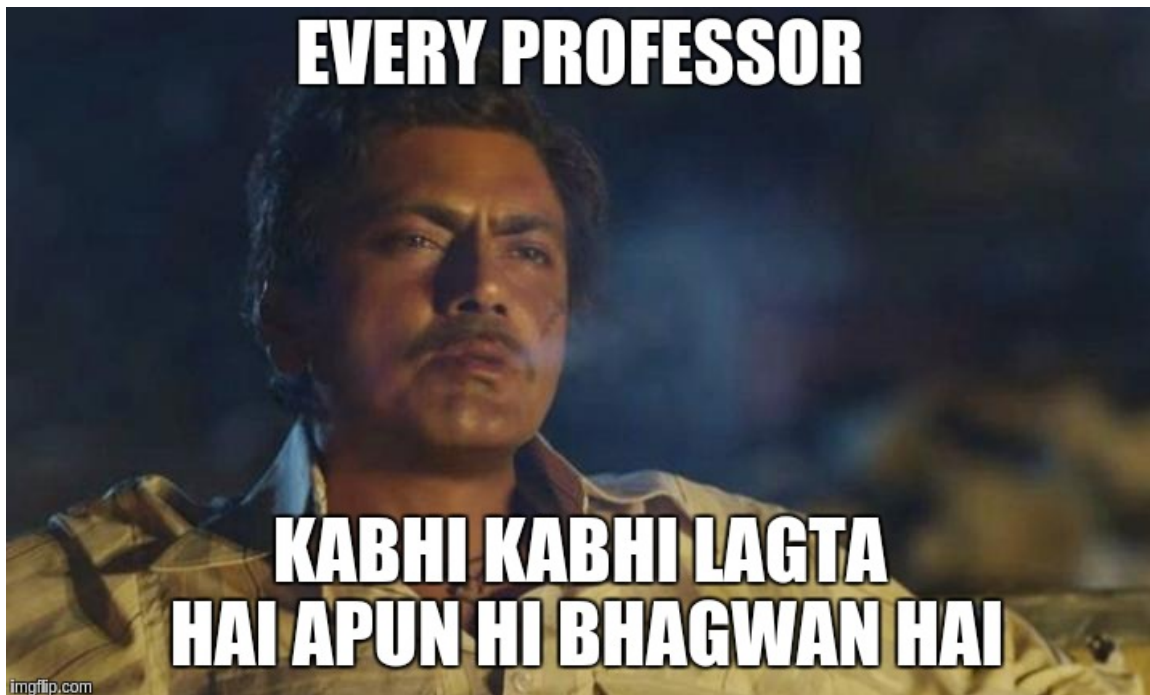
1. Length of the song in front of each song.
2. Drag & drop feature for inserting the playlist.
3. Rating for each song.
4. Equalizer.
5. Play music via internet.
6. Should play audio from the video songs.
7. More points for attractive UI.
8. Fetch lyrics online.

Professor Hunt

Description:

“Kabhi kabhi lagta hai apun hi bhagwaan hai” must be the saying of many professors and even the Guest faculty when they realize they rule over us. That’s why “Desh sankat mai hai!” isn’t it? In your ears only one eternal voice must be reverberating, “Kabhi kabhi lagta tha mukti dedu isko!!” To simulate the above, help Gaitonde to make the perfect Professor Hunt game ever.

P.S: Think of the duck hunt video game!!



Basic features:

1. There are two game mode :- 1 duck and 2 duck. Depending on the mode selected, one or two ducks will appear at a time.
2. Each round tasks the player with killing the series of ducks by clicking them on screen.
3. Each time a duck (or pair of ducks) flies on screen, it will remain in air for fix amount of time. If the player misses the shot, the ducks will fly away.

4. To progress to the next round, the player must shoot down a set number of ducks, with the number of ducks required to progress increasing as the player proceeds further into the next round.
5. Should the player fail to shoot down any ducks in a set, the dog will rise out of the bush and laugh at the player before the next duck appear on screen.
6. The dog will also laugh at the player when a Game Over occurs by failing to shoot enough ducks to proceed to the next round.

Advanced features:.

1. Add third game mode, Clay Shooting, which is a fairly accurate representation of the real-life sport, featuring pairs of clay pigeons that must be shot as they are launched into the air, becoming smaller and harder to hit as they recede further into the distance.
2. Add different difficulty levels based on speed of flying, flying time of ducks in air, the flying path of ducks etc. (choice should be in random order)
3. In the bottom of screen show statistics (remaining ducks available, score of player, round no. etc.). Show not only just numbers. Be creative.
4. Store player's information with his overall statistics and he should be able to view his progress in the form of chart or graphs.
5. Add more sound and animation to grab more points.
6. An impressive UI will grab more points.

Brother's Eye

Description:

Felicity Smoak is in search of a new programmer who can help team Arrow track down criminals. She came up with an idea of creating a program - Brother's Eye, which will keep an eye over all the prospective candidates by looking at their various coding profiles. But it's just the beginning, she has bigger plans for Brother's Eye. Being an open source enthusiast she thinks that this program will also help open source communities and hence she doesn't want to make it Queen Enterprises proprietary. Help Felicity to develop Brother's Eye.



Basic Features:

1. User sign up and login.
2. User can connect their profiles at various programming sites like Spoj, CodeChef, GitHub, StackOverflow etc.
3. A global leaderboard of all registered user based on stats received from all the programming sites.
4. A user home screen where user can see various trending coding problems, repositories, question and articles all at one place. (You are free to explore various API's to achieve this).

5. A list of contests on various programming websites with an option to add a reminder half an hour before contest starts.

Advanced Features:

1. User can follow other users to see their activities on various programming websites.
2. User can search for problems, repositories and other stuff based on tags.
3. A user dashboard which will represent the user's overall performance on all programming sites. (For eg. In case of competitive coding websites you can show overall accuracy, sitewise accuracy, streaks, contest rating graphs and calendar of submission. As always more statistics more points)

Open Problem

Description:

Professor X is organising an contest for those who are completely new to the world of software development. As he knows that as a beginner it is very difficult to develop solutions to complex real life solutions, that's why he has asked all his participants to choose a problem according to their own ease and develop a working software for the same.

PS. Professor X has declared that the one who wins this contest will get admission to Xavier's School Of Gifted Youngsters.

