



Softablitz Problem Statements

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

Facebook group:-

<https://www.facebook.com/groups/45494592821459>

Contact Us:

Abhishek
Kumar Singh

Surbhit
Awasthi

Shreyas
Dobhal

Ashish
Hattimare

Paras Saini

Himanshu
Purwar

Sri Ram
Khandelwal

Tushar
Srivastava

Shreshtha
Dwivedi

Araish Aftab

Basic Instructions

1. A team can consist of at most 3 members.
2. All branches are allowed, but each team must have at least 1 member from either CSE or IT.
3. Plagiarism in any form will result in elimination.
4. Teams can attempt at most 2 problem statements.
5. Each problem is of certain points (Points will be added later). Full points will be awarded only when the features described in the problem statement are correctly implemented.
6. Each problem has a specific theme, bonus points will be awarded if the final solution is in accordance with the theme.
7. Bonus points will also be awarded if the team implements features other than the one specified.
8. Try using git and GitHub, and remember to keep your repositories private.

Few Pieces of Advice

1. Firstly read each and every problem statement and then decide which one suits best with your current skill set. Don't just run after points.
2. Try to build a complete solution to 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 a very important part of any desktop application but first try to implement as many features as you can. In case you get 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. Remember to keep the spirit of Softablitz ablaze by not using libraries to implement features unless specified to do so, any libraries/packages/JARs used which are not mentioned by us will lead to negative points.
6. Last but not the least for any doubt regarding problem statements contact coordinators as we are always here to help.

This3Bhushan

Description:

Kartik has a huge collection of movies, songs, study material and books on his laptop. He wants to share this collection with everyone, especially with his juniors so that they have ample sources to study from and enough movies and songs to quench their entertainment needs. He conveyed his wishes to his friend Amit Jain. Amit Jain being busy in conducting Inception but a true friend wants to solve this problem by creating a file-sharing/streaming software. But since he is busy he wants your help.

Inspired by torrents Amit Jain wants to develop a distributed file-sharing desktop application which stores files chunk by chunk on multiple nodes so as to increase the availability and accessibility.

Basic Features:

1. A user account to which each user can log into and create his profile and can share content with others.
2. A dashboard for the user showing the files he has downloaded or shared, the number of people who have downloaded the files he/she has shared.
3. While sharing a file the user should be able to define what kind of file it is. (Eg: whether it is a movie, a song video, a book, a research paper etc.)
4. A file shared by any user should be stored on multiple nodes either as a complete copy or in the form partial chunks. (i.e. in a Distributed Manner)
5. The chunks of any single file must be stored in at least 3 different nodes.
6. A user should be able to download a file using a unique identifier (i.e. a file (or any such entity or object) which exclusively defines that particular file which the user wants to download).

7. If identical files are shared by different users, they should have the same identifier.
8. Create an interface where the user can search for different kinds of files (songs, movies, documents etc. and remember more creative fetch more points)
9. Create a user interface where users can search a file using the file title/name.

Advanced Features:

1. File shared by the user can have multiple tags like a movie shared by a user can have tags like action, fantasy etc., a book shared by a user can have tags like ML, coursebook etc.
2. Create a search utility where the user can view files and search files using various other parameters like file type (Document, Video, Song, Movie etc.), tags etc.
3. For a file (or a part of a file) which is stored on multiple nodes, downloads for that file should occur parallelly from all those nodes which are online and which have a part of that file.
4. In case of multimedia files like videos or movies, play the video/movie without downloading the whole file. (You can use 'vlc' or similar library).
5. Recommend multimedia files for users based on their download/view history. (You can use external libraries for this.)
6. If for a search query there are multiple results, rank the results based on the availability of that file on different nodes.
7. A user who has still not downloaded the whole file (but has some part of it) can have other users downloading that part of the file from his node.
8. Bonus Features: Points for UI which follow Material design conventions.

Sakshatkar

Description :

Sakshatkar is a video chat application (inspired by Skype). This application will help individuals to help people overpower the monopoly of private brands like Skype by providing an alternative with similar features in an open-source project.

Basic Features :

1. User must be able to login or sign up to the system.
2. One must be able to search for other users and view their profile.
3. Profile page for each user (profile status, username , profile photo, company, contacts etc.)
4. User must be able to conduct a video chat with another user (video + sound)
5. Users should have the ability to mute the microphone.
6. Users should be able to chat with each other by sending text messages (like conversation window next to the video)
7. One can send files to other users.
8. After the call ends feedback should be collected on call quality

Advanced Features :

1. User must be able to conduct voice calls with another user.
2. Show notifications for calls and unread messages.
3. You must be able to track all the users that are currently online
4. A user can set his status as Available, In Meeting, or Busy. Indicator for online or offline must work automatically.
5. Support for friend request before video calling another person.
6. If a user is offline others can see his last seen.
7. One should be able to send messages even to those users who are currently offline.
8. The chat history should be maintained (Messages should not be lost when the application is reopened i.e. local backup).

9. One should be able to share his Desktop screen to others who are connected together (View only).
10. Friend suggestion to a user based upon “friend of a friend” ontology.
11. Group video chatting.

Note: Additional points will be awarded for synchronising audio with video and following conventions of material design.

For this project, you can use

- OpenCV
- Webcam Capture API (<https://github.com/sarxos/webcam-capture>)
- Xuggler

InterCode

Description :

InterCode is an application that takes Telephonic Technical Interviews to next level, by allowing Interviewer and Interviewee to work on the same Code Editor window simultaneously and be able to compile and run the code to check the correctness.

Basic Features :

1. Both must be able to communicate with each other by sending and receiving messages (in a separate conversation tab).
2. The interviewer can enter details about the candidate in the database.
3. They both must be able to work on the same editor window simultaneously (typing in one directly changes the other's code).
4. Both should be able to compile and run C, C++, Java, Python codes.
5. The application must precisely display compile and runtime errors and output for a given set of inputs.
6. Implement options like Find and Replace (provide an option for regex)
7. Keyword highlight according to the language selected.
8. Change font size and style in editor.
9. Show line number in the editor.
10. The interviewer must be able to give a question to the Interviewee for him to solve.
11. After interview Interviewer can rate and score the student and the result is saved.
12. Fetch details of best of all the candidates.

Advanced Features :

1. The interviewer should be provided with a timer to keep a track of the solving speed of the student.
2. The interviewer can write detailed notes regarding the performance of the student (timer and notes visible to interviewer only).

3. In case the candidate tries to cheat by opening another window (of another application may be a browser) then notify the interviewer and send the screenshot of candidates' desktop to him.
4. Provide Auto-indentation for the code.
5. Auto-complete should be possible which suggests a list of possible words when ctrl+space is entered. (Tri to optimise)
6. Handling synchronization and preventing inconsistency will help you get extra points.

UNO™ Multiplayer

Description :

UNO is the classic card game that's easy to pick up and impossible to put down! Players take turns matching a card in their hand with the current card shown on top of the deck either by colour or number. Special action cards deliver game-changing moments as they help you defeat your opponents. These include Skips, Reverses, Draw Twos, Wild and Draw Four Wild cards.

Basic Features :

1. Users on different systems should be able to connect to a game hosted by one of the users (min 2 users).
2. Deck of UNO cards should be shuffled and dealt to the players.
3. Design an impressive UI.
4. Implement the standard rules of UNO™ (You may or may not implement the in-house rules.). Refer this link for standard rules https://service.mattel.com/instruction_sheets/42001pr.pdf
5. Users play their move turn wise (either in clockwise or anticlockwise manner which can be changed by 'Reverse card')
6. One player can only see his own deck of cards and to him all other players' cards seem face down.
7. A central pile of cards is maintained with one face up card with which the match is to be made for a valid move (except 'Wild' and 'Draw +4')

Advanced Features :

1. If a player is left with only two cards, activate a 'Call UNO' feature when he plays a card, and activate a 'CAUGHT' feature for all the

other players after the former player has played the card. Current player has to press the 'Call UNO' button before anyone else clicks on 'CAUGHT' button, failure to do so will result in a penalty of 4 cards

2. More than one cards can be dropped in a single move if their face value matches (eg. On a Blue 4, one Green 4 and one Red 4 can be played)
3. In case a user has an identical card, same as that on the pile, that particular user can play out of their turn (before the user with current turn plays) and the game will continue from him (your initial deck must have face cards with multiple copies of the same value)
4. If someone throws a Blank card of any color then every player shifts their playing deck to the next player (i.e player 1 will give his cards to player 2, player 2 will give it to player 3 and so on ..)

/dev/pts/0

Description :

A computer terminal is an electronic or electromechanical hardware device(in olden times) that is used for entering data into, and displaying or printing data from, a computer or a computing system. The teletype was an example of an early day hardcopy terminal and hence the name tty which stands for teletypewriter.

Features marked with (-) means:

- Don't use the 'system()' function call to execute commands or make processes using ProcessBuilder or any other thing, that will not be counted. Implement functionality of commands using your own logic and using the language's system library.

Basic Features :

1. A user can log in to the terminal.
2. Prompt character turns red if the last command exits with non-zero code.
3. Implement basic file management commands like ls, mkdir, rmdir, cd, exit etc. (-)
4. Implement multiple basic system commands, like for OS name, displaying, IP Address, router table etc. (-)
5. Create a command to set system region-wide proxy (regions can be specified with different options).
6. Add history feature.

Advanced Features :

1. Show current folder stats like memory occupied, blocks, number of files etc. (-)

2. Current git branch status. You can use any symbol you want to indicate different stages in a git repository. Different stages can be like untracked changes, uncommitted changes in the index, unstaged changes, renamed files, stashed changes, unmerged changes, ahead of the remote branch, behind remote branch etc.
3. Coloured output or themed terminal will get more points.
4. Provision for path variable extension.
5. Algorithmic autocomplete.

Note: Implementation of all commands is not a compulsion. Implement as much as you can and some good implementations will get bonus points.

GitFlix

Description :

GitHub has been the largest collaborative version control site for years. Until now, presenting before you is GitFlix a git-based version control desktop app. GitFlix will basically help collaborators work on team projects while maintaining track of every change being made.

Basic Features :

1. Allow user to set up basic configurations like username, email, etc.
2. User should have the ability to create a repository, add a repository, or clone a repository.
3. Setting up different remotes.
4. User can commit changes and push their work on a remote repository. Users can also pull changes from the same.
5. User should be able to create and manage different branches.
6. User is able to keep track of files in different stages of the git workflow.
7. User can view their files (images, text files) in an editor.
8. Users are able to find differences between two files, or across two commits.
9. User can see commit history on the branch they are currently on.

Advanced Features :

1. Create a well-designed UI so that the user is intuitively informed of the different states of files in the git workflow. (Eg: If a new file is created in the repository the UI should show that this file is untracked, or if a file is edited the UI should show the file is unstaged: modified).
2. User should be able to resolve merge conflicts in the editor
3. Users should be able to move to and from previous commits in the workflow using the UI components.
4. User should have the ability to rename and delete a branch.

5. Show the graphical representation of git commit history of the current branch and the whole repository (i.e Network Graph).
6. User should be able to stash the dirty (uncommitted) work.

NEWSGyan

Description :

NEWSGyan is a desktop application that helps one stay connected to the world by keeping clients up-to-date with the daily headlines.

Basic Features :

1. Login/Register for the new user.
2. Fetch NEWS articles from multiple trusted websites.
3. Ask the users about their preferences at the time of signup (may be updated later as well)
4. News articles must be displayed based on the preferences of the user (sports, politics, world.)
5. Each article content should be concise and limited (show first few lines or so (be creative))
6. Articles can be searched based on their category.
7. User can click on these articles to read them in further detail
8. Weather report and currency exchange rates can be shown as well.
9. UI must be impressive and creative (try making different tabs, views and cards to categorize articles).

Advanced Features :

1. Widget for the application that always stays visible on desktop and shows Top HeadLines (redirects to the application when clicked).
2. Upvote or Downvote an article, and based on these arrange them in sorted order under their respective category (don't forget to deal with new articles that have zero total votes moreover default articles can be shown using this count).
3. Some articles can have images as well (be creative and add your custom image if there is none provided).

4. User should be able to save their favourite articles locally and view them any time.
5. Create a 'recommended for you' section where you can show articles from websites whose past posts the user has upvoted.

Bob the Destroyer

Description :

You are required to build a simple gaming application which is similar to the traditional 'Brick Breaker' game provided in Miniclip and Google Play Store.

"Better Game = Better UI + Better Algorithm"

Basic Features :

1. At beginning default profile should appear, user can play as a Guest or create his own profile. And should be able to switch profiles.
2. Should provide help to the player to understand the game and its components.
3. The player should have the option to play with different modes
 - i) Allow the player to play only one level at a time.
 - ii) Allow the player to play the game from level 1 to the end.
4. The application will show to show the top scorers.
5. The game should consist of a slider which is either controlled via mouse or keyboard (or both and can be changed with player preference from the setting).
6. Provide different sound effects to the game to enhance the player's interest.
7. After breaking the brick, the ball should reflect in the opposite direction.

Advanced Features :

1. Try to implement movable bricks in the game.
2. Try to provide different power-ups in the game (More power-ups gives more points to the project).
3. The game should consist of multiple levels, and difficulty gradually increases as the game progresses.

4. Allow the player to have some sort of 'God Mode' which allows the player to activate any powerup at their own will.
5. Try to reflect the ball in a realistic direction. For example, after the ball hits the corner of the slider or the slider, it may reflect back at a different angle.
6. User should be able to track his performance and statistics. Be creative show the information in the form of graphs.
7. Add the concept of achievements and medals, which will be unlocked based on certain tasks (like stay idle for 5sec, and many more).
8. User should be able to see all his unlocked medals and also see how to unlock the locked medals.

“Try to add extra new features to the game to make it more challenging and fun. Be creative with the approach”

Sudoku X

Description :

Remember the old classical Sudoku on those newspapers. Well, that has evolved to next-level, the Sudoku X. It provides interactive features like multiplayer time-based mode and auto solving

Basic Features :

1. Implementation of basic rules of sudoku.
2. Setting a solvable puzzle.
3. Selecting difficulty level Easy (2x2), Medium (3x3), Hard (4x4).
4. The player must be timed properly.
5. Check to disable the user from entering invalid input.
6. A different mode in which the user can enter an unsolved puzzle and see the solved answer of the puzzle.

Advanced Features :

1. Multiplayer mode(n different systems) in which every player is given the same puzzle to solve, and a leaderboard is generated based on the order they solve it.
2. The sudoku generated should be proper sudoku(only one possible solution).
3. Suggesting the possible values that can be entered at a particular cell.
4. Provide the option of hint which fills a number at a random cell (and penalize for the use of Hint as well).
5. Add features like Undo and Redo of moves.
6. Varied difficulty level even in a particular difficulty mode itself.ve show the information in the form of graphs.
9. Add the concept of achievements and medals, which will be unlocked based on certain tasks (like stay idle for 5sec, and many more).
10. User should be able to see all his unlocked medals and also see how to unlock the locked medals.