TEAM 10 : Deadlock and Synchronisation (Website)

Technologies Used:

- HTML
- CSS

  For design of webpages for diiferent algorithms
- JAVASCRIPT (for event management like Onclick)
- JQUERY (for transition effects and animation)
- BOOTSTRAP (framework for HTML, CSS, JS)

SOFTWARES TO BE INSTALLED:

- Any text editor like notepad, wordpad etc are sufficient.
- Any code editor like VSCode, sublime etc.

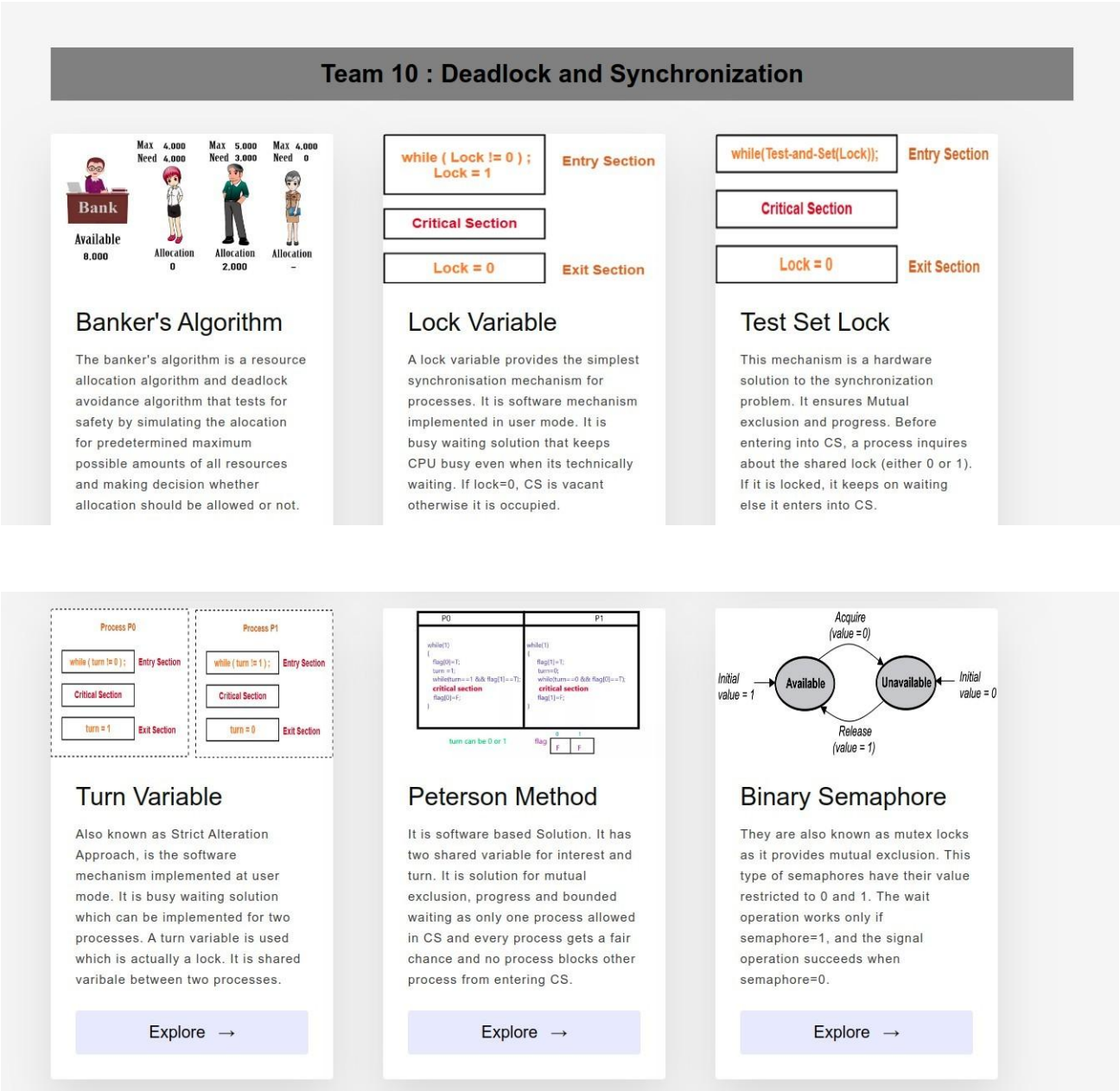
  Download links :
  - VSCode : https://code.visualstudio.com/Download
  - Sublime : https://www.sublimetext.com/3
- Trick in VSCode : After installing VSCode, download live server extension and right click on the file(for eg. Index.html) and select open with live server option and it will be opened in your default browser and any changes you will make in the file it will automatically reflected live in the file opened in browser.

PLATFORM DEPENDENCIES:

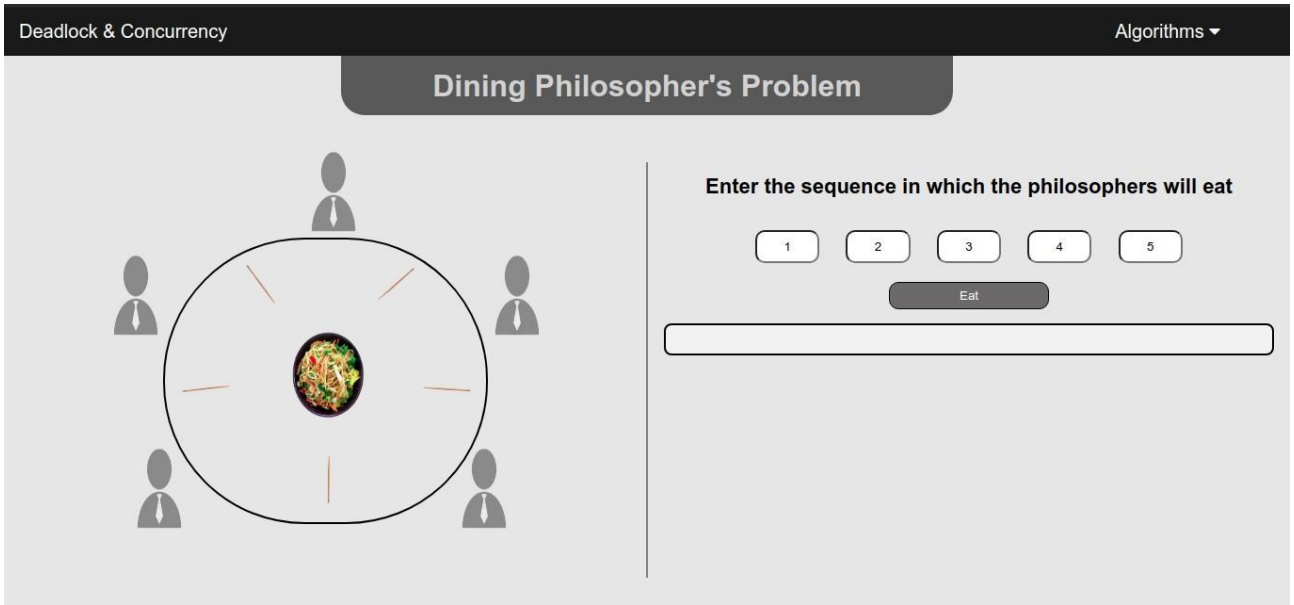
- There are no such platform dependencies as so. But using code editor will make the work easier to unserstand the code than to look it in the notepad.

HOW TO RUN THE PROGRAM:

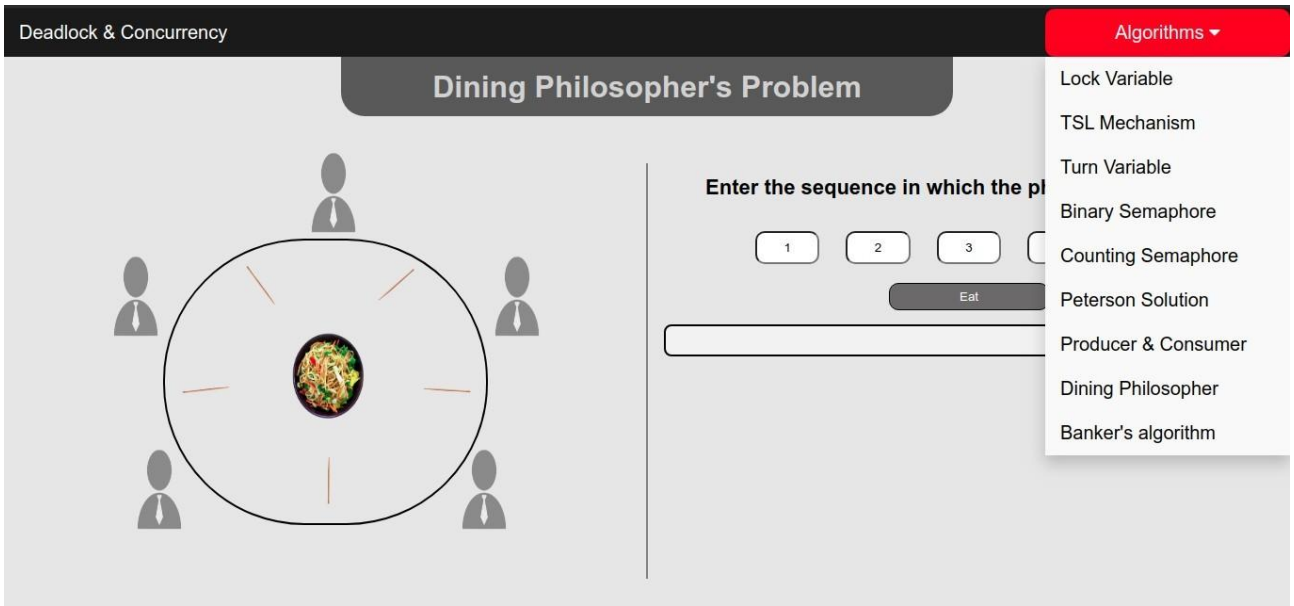
- As soon as you download the .zip (or .rar) file of the codes, extract the codes and to open the Home page, open "home.html".
- As we opened home.html, below screen will be visible...



- Now above images are of home screen which have algorithm's brief information about the task it performs and how it performs. Clicking explore will lead you to the the algorithm page of the particular algorithm. For e.g. we have clicked explore button of Dining Philosopher's algorithm, than the below page will open...



- Now this is the home-page for Dining Philosopher's algorithm and on top of every algorithm home page there will be navigation bar which have two options:
  - Deadlock & Concurrency
  - Algorithms
- On clicking Deadlock & Concurrency button, it will lead directly to the home page of our topic that is shown in above images.
- On Hovering over Algorithms button, we will get list of all algorithms that are covered under the topic. The user can click on any of the algorithm he wish and he will be redirected to the home page of that algorithm as shown below.

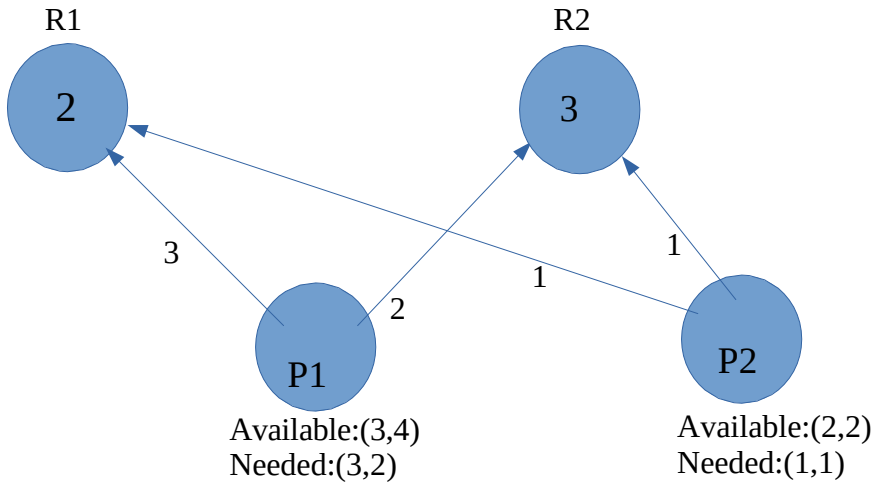


ENHANCEMENTS THAT CAN BE MADE:

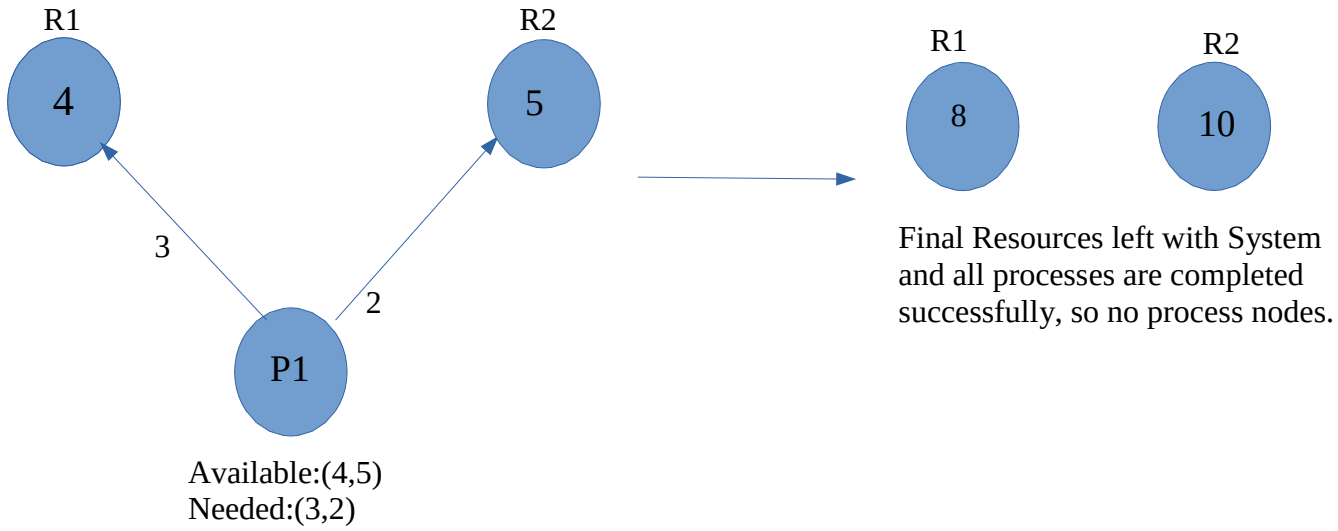
There can be certain changes that can be made to certain algorithms.

1) Producer and Consumer algorithm: We have implemented this algorithm statically. That is we ask user to enter the buffer size and after submitting we have created that many blocks. If user clicks producer than it will fill a block with item and if consumer than it will consume a block item. And if buffer is filled or emptied completely than it will prompt user accordingly. The change that can be implement is automate the producing and consuming activity. That is user doesn't have to click the buttons, if buffer is emptied than it will fill until full and vice-vers for consumer.

2) Banker's Algorithm: The animation can be improved further. The user can add nodes for processes and resources(node value represents reources available) respectively. Represent edges connecting nodes as the process needing the resources and weight of the edge as the resource needed. If process completes it execution, remove process from the graph and add the free resources to particular resource node. For e.g.



Now as seen in figure Process P2 can be executed first as it needs (1,1) resources and available are (2,3). So removing P2 and adding resources available with P2 to total resources.



So likewise we can show graph stepwise.

3) Some General enhancements that can be applied to both binary and counting semaphores, TSL, Lock variable, Peterson Method, Turn Variable is that for some processes and resource available we can create comparison chart for all algorithms that shows that will algorithm can be used and what results will be yield at the end.