**N in a row-**

This is a web game based on the game "4 in a row". In this game the board and the goal are not fixed.

In addition, you have 3 types of games:  
Regular – like the original game "4 in a Row ".  
Circular – you can win by create a sequence in a circular form from left to right and from top to bottom.  
Popout – Pop out your disk from the bottom.

**General explanation of system design:**

In the game implementation we separated the logic (the "GameEngine" module) and the user side ("webApplication" module) as required.

To connect the two parts above, we used the class "ServletsUtils" as demonstrated in the class in the final example.

**The "webApplication" module:**

This module represents the client side. Is responsible for the user interface and operation of the "engine" game.

This module is responsible for receiving the input from the user -> activating the logic object accordingly -> and displaying the relevant output as a result of the input command.

This module combines the following tools: servlets, javaScript (jQuery, ajax), jsp, html, css.

On screens 2,3 we perform information update automatically for each player (each browser) in the form of Pull as we have learned in the class and according to the update requirement is performed every 2 seconds.

This update is performed by using the capabilities of ajax, jquery, javaScript and receiving information from servlets in the form of a json object.

The display of information is done by using html, jsp and css.

**The GameEngine-**

This module is responsible for the game engine. In this section the logic of the game is realized.

You can find an object named RoomsController that manages the list of available game rooms and the list of entrants connected to the system. In each room there is an object that is the game engine for that room which includes:

The structure of the clipboard and all the relevant commands for the data structure that saves the board,

The list of players in the current room and the game engine itself, depending on the type of game played.

In each engine you can find all the logic of the game itself, checking line, saving information, tests related to the logic (full column) and so on. In the realization of the engine, polymorphism is used so that we send commands to the engine without the need to know its type at a given time.

**How to open the app:**

After uploading the Tomcat server and uploading the NiasRow.war file, please go to the following address:

http://localhost:8080/NinasRow /

**How to play:**

For playing the game one of the users need to open a game room. You can open a game room by uploading one of the xml files in the folder "xml files". The game will be active when the room is full.