# Report

## Live demo
http://testforcoolgames.co.nf/

## Source code
https://github.com/vtsitlak/bestgames

## Technologies used:
 HTML5, Less, Bootstrap, JavaScript, AngularJS, jQuery, npm, bower, gulp

## Installation:
npm install

gulp

## Development:
All source files, JavaScript, AngularJS, images, HTML5 are on src folder. After installation a public folder will created containing all the ready files for the project. Npm installs gulp and several gulp plugins that helping with the development. Bower installs libraries and packages for the project. When running gulp, several tasks are running and extracts the final project on Public folder.

## Gulp tasks
- Bower: installs all bower items from bower json file to bower_components folder
- Fonts:  move fonts to public folder
- Htaccess: moves  .htaccess file to public folder
- Html: minify and moves all html files to public folder
- Json: task minify and moves json file to public folder
- Scripts: concatenate, minify and moves JS files to public folder
- Less: compiles to css, concatenate  and minify less files
- Css: task concatenate the previous less compiled file with the rest css files from several packages and libraries, minify them and moves the final result to public folder as main.min.css
- Img: optimizes the image files and moves them to the public folder
- Watch: task watch for changes on files and updates the files, is useful while in production, you can comment out this task so that gulp task stops run.

## Bower components
- jQuery: The library for jQuery
- Angular: The library for AngularJS (1.5.8)
- Bootstrap: The library for Bootstrap
- Font-awesome:  Font awesome package
- Angular-Bootstrap : Native AngularJS (Angular) directives for Bootstrap
- Angular-Animate: AngularJS animation module
- Angular-ui-router: he de-facto solution to flexible routing with nested views in AngularJS
- Angular touch:  An angular module to add directives for touch devices
- Angular-metatags: Module for providing dynamic Meta Tags to Angular route

- angular-rating: a module to help you create a rating with some cusomizable options
- angular-sanitize: AngularJS module for sanitizing HTML

## Project structure

- Root: On root there the bowe json file with the libraries and packages, the package json file with the plugins for the gulp tasks, the gulpfile.js that create the gulp tasks and the src folder where all the files for the development are.
- SRC folder:
  - HTML folder: there is index.html file
    - parts sub folder: the templates for the rest of the pages and angular directives
  - Less folder: we use the main.less file to import all less files
  - Fonts folder: the fonts
  - JS folder: A subfolder for AngularJS scripts.
    - Angular folder: there is the app.js for the module and the routes and the following subfolders for the controllers services and directives
      - Controllers
      - services
      - Directives
    - jQuery folder: contains the flagstrap package

## Main projects functionalities

- Design: A fully responsive mobile first web site
- Site optimization: All html, JavaScript, css, images and json files are minified and optimized using the gulp tasks
- Carousel: Using angular-bootstrap-ui carousel with left and right control. Followed by the, sanitized HTML, game description on the bottom of the carousel and a set 0f 3t different colors buttons for play, play later and favorite. Favorite button has a toggle functionality to change the hear icon color on click
- gameItem: A directive for an game item template. Contains a game icon or a default icon in case of broken link, the start rating, the name and tooltip with the name
- Routing: Using the angular-ui-router and set the html5 mode to true to avoid the '#' symbol. Needs to manually edit the .htaccess file to set the index.html location, and also define the base href location on index.html file, depending on server structure
- SEO : Added description, author and keyword metatags. When we create dynamic a new state for the game links, we add the game dynamic to these pages the name, seoKeywords, description, categoryIds, tags and id of the game, as they are on json files. To add dynamic these metatags we use the Angular-metatags module, just after we create dynamic a new state, on stateService. We also add the same texts on hidden elements on Carousel and on gameItem. Dynamic search on the articles by using keywords
- Link to games: The games are loading inside the page, by creating dynamic a new state when the user clicks on play button, or on gameItem icon. Using the json data about the game we call the stateService and creating there the new state, if not already exist.

- IMPORTANT! To prevent the No 'Access-Control-Allow-Origin' header error has been used a CORS proxy, the game url link is 'http://cors.io/?' + game link . Because of the limitations of the CORS proxy, and the limitations on the free server the game links often are not working, or working very slowly. Also many links having a wrong key messageTherre is the option to coment out the addState function on mainContoller and uncomment the window.location.replace function to just go to the external link of the game