



Puspresnas
Pusat Prestasi Nasional



Member Of
worldskills

KISI-KISI

**LOMBA KOMPETENSI SISWA (LKS)-SMK
TINGKAT NASIONAL XXX TAHUN 2022**

BIDANG LOMBA

**Teknik Desain Laman
(Web Technologies)**



Teknologi Informasi & Komunikasi

SPEEDTEST MODULE

Part A: Website Design

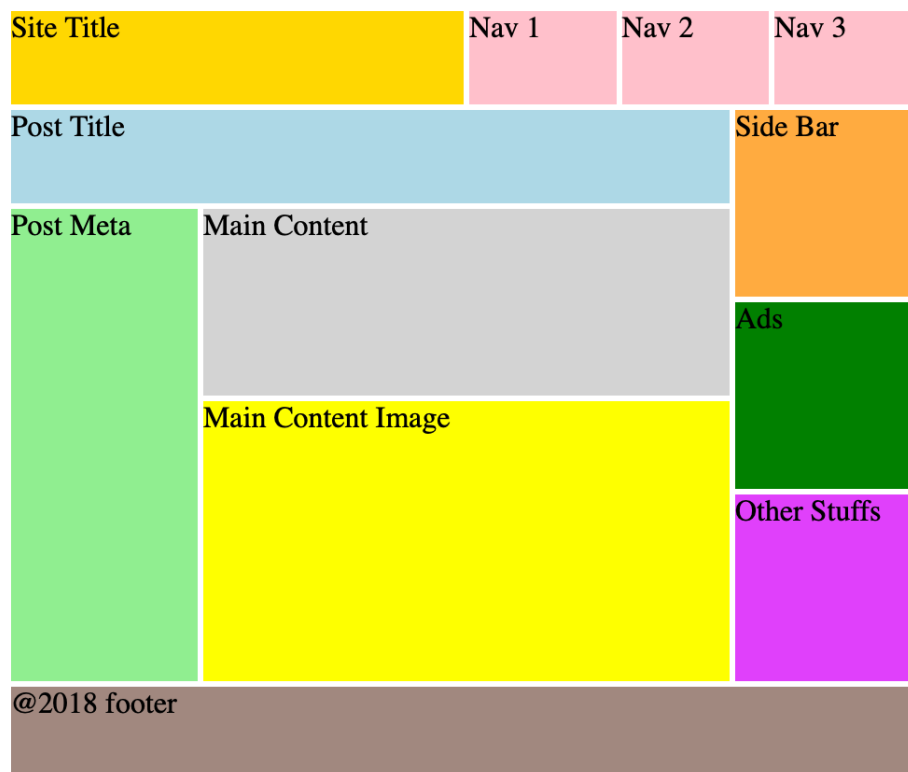
UI-kit using design tools

Create a UI-kit of the following elements: text box, button, drop-down list, checkbox, radio button, label with checkbox, label with radio button, drop-down calendar input, range, progress bar

Part B: Layout

CSS Grid

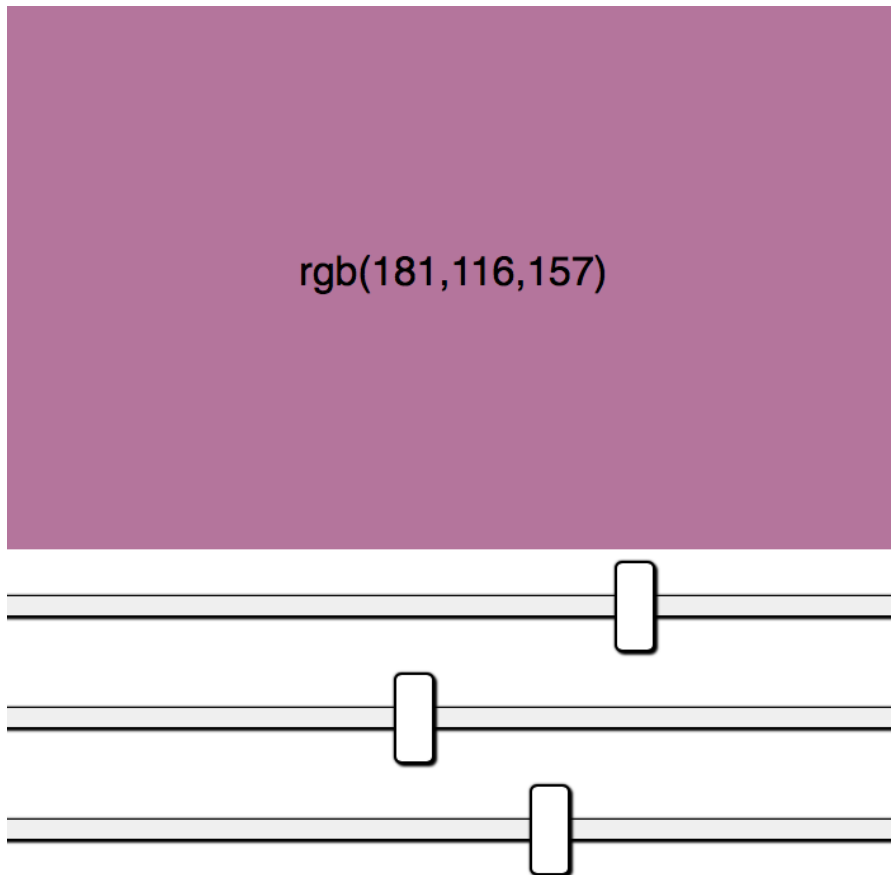
Please implement the following layout in CSS grid.



Part C: Front-End Development

RGB Slider

Please implement an RGB slider as following. There are 3 sliders for adjusting values of red, green and blue.



Part D: Back-End Development

PHP Verification captcha

Please use PHP to produce verification code image (captcha), which must comply the following rules (please see the provided images):

- a. It randomly generates four characters, consisting of English Alphabet (A-Z) or digit numbers (0-9).
- b. The four characters must be slightly rotated.
- c. The four characters must not be located at the same row.
- d. The image contains at least 3 random lines.
- e. At least 3 mixed noise points.

Part E: Content Management Systems

CMS Administration

Set up a wordpress site with several credential listed below:

1. Username: admin
Password: LKSN2020
Role: admin
2. Username: irfan_LKS
Password: irf4n_Gan\$
Role: editor

Instruction for Competitors

Save your files in your root directory on the server called "module_speedtest". Inside that folder create a folder for each part (A-E).

CMS MODULE

Introduction

You are a freelance worker working in the field of Web Technology and you have been asked to develop a new website for COVID-19. Your client has heard that you are good at building content management systems.

Description of Project and Tasks

This module involves knowledge about website design, website layout techniques, client-side scripting, and server-side scripting, all combined in one CMS project. You will be using one of the most popular content management systems - WordPress.

Your job is to develop a website backed by a content management system. Please create your own theme as a child-theme of a given starter BlankSlate theme. Please name your theme **COVID19_Indonesia**.

The purpose of this website is to display the latest news about COVID-19 in Indonesia as well as selected news stories from selected editors. A newsletter can have a list of upcoming events. The purpose of this website is to provide information and news regarding COVID-19, from general news, socialization and urgency of conditions in Indonesia and to provide a list of upcoming events that can be followed for people who wish to participate. The target audience for this page is all elements of society in Indonesia. This can vary widely, therefore you will determine your interpretation of the target audience (based on "Indonesian society"). Also make sure you have the headers and slogans defined in the content management system.

Admins and editors should have the possibility to add and modify each entry for each story either to update existing information or to add new news to the list. The site is accessible for visually impaired visitors.

CONTENT MANAGEMENT SYSTEM

For administration tasks, we need two user profiles, Admin and Editor:

- The Admin user - access to the complete WordPress main dashboard.
 - Username: admin20
 - Password: 20admin
- The Editor user – access as editor role in the CMS.
 - Username: 20editor
 - Password: editor20

We want the CMS login page to be white-label. That means the login page should not show the CMS default logo and add all the editor's logos.

Also the login page should not include any “wp” wordings for white-label reasons. Please make the backend admin URL as following: <host>/admin/

MANAGING NEWS AND PAGES

The client wants to be able to add, edit, update, delete news records.

NEWS BLOG POST

From time to time, there will be news from the website as well as from specific editors. These blog posts should be categorized.

The client wants to show news posts from all categories in the home page. While in the specific editor page, the client wants to show news posts from only the category for the news.

Post Categories

- News Updates
- COVID-19 Events
- Each selected editor

PAGES AND URL STRATEGY

Accessibility is very important to our website. We would like to leverage different levels of accessibility optimization.

URL level of accessibility

All editors pages should have their own URL permalink.

News posts should have following URL structure:

- All news: <host>/news/
- News from a category: <host>/news/<category name>

DASHBOARD CONFIGURATION

The client wants to have At a Glance, Activity, Quick Draft in the dashboard. Please configure these dashboard widgets and remove all others in the dashboard.

MENUS

The menu should stick to the top when scrolling down the page. The menu has the following items.

- Home
- Our Category (List from categories)

FRONT-END REQUIREMENT

You will need to create your own theme, based on the given starter themes. You will need to implement the following elements for your web page:

- A footer with copyrights and social media links
 - “Copyright © 2019 - All rights reserved” where the year should be the current year dynamically based on server time.
 - Footer plugins should be used for footer links to Twitter, Facebook, and Instagram. User can change the social links.

Feel free to add and change as many elements as you like, but the elements in the list need to be present and your design should blend in with the given design.

The client does not like refreshing the whole page. Page transition should be loaded via JavaScript with animated transition effect. (This does not apply to admin area)

HOME PAGE LAYOUT COLUMNS STRATEGY AND MOCKUP

The layout should be responsive for different screens including mobile and desktop.

Home page on the desktop screen should have a multiple columns layout. Left half is some kind of cover and images and the right half is the news.

Home page on the mobile screen should have a top-down layout strategy. Each column takes the full width of the screen, except for the list of selected editors. They should have a left and right 2 columns layout.

FURTHER OPTIMIZATION TASKS

The page needs to be optimized for search engines.

You can choose one of the provided plugins or implement your own SEO.

Remember to optimize certain parts of your website (e.g. urls, sitemap, ...).

Secure your page by installing and configuring a security plugin.

Note: If the plugin you would like to choose is broken or does not work as expected, you need to choose another one or feel free to fix this plugin.

INSTRUCTIONS TO THE COMPETITOR

The page should follow the Web Content Accessibility Guide.

For the working folder you need to name your folder to “**module_cms**”.

EQUIPMENT, MACHINERY, INSTALLATIONS, AND MATERIALS REQUIRED

ITEM	description
advanced-custom-fields.5.8.0.zip	Plugin
all-in-one-wp-security-and-firewall.zip	Plugin
autodescription.3.2.4.zip	Plugin
blankslate.2019.1.zip	Theme
contact-form-7.5.1.3.zip	Plugin
contact-form-plugin.4.1.5.zip	Plugin
contact-forms-builder.1.5.2.zip	Plugin
elegant-responsive-content-slider.1.0.1.zip	Plugin
jetpack-6.9.zip	Plugin
sample_underscores.zip	Theme
seo-by-rank-math.1.0.24.zip	Plugin
wonderplugin-carousel-free.zip	Plugin
wordfence.7.3.2.zip	Plugin
wordpress-seo.11.2.1.zip	Plugin
wp-responsive-jquery-slider.1.7.zip	Plugin

MARKING SCHEME

Section	Criterion	Judgement marks	Measurement marks	Total
B1	CMS Theme	0	3.0	3.0
B2	CMS Configuration	0	3.75	3.75
B3	CMS Design	1.5	1.5	3
B4	CMS Layout	3.0	3.5	6.5
B5	CMS Category	0.5	3.25	3.75
B6	CMS Admin	0.25	2.25	2.5

MODULE SERVER

CONTENTS

This module has the following files:

- MODULE_SERVER.doc
- MODULE_SERVER_MEDIA.zip

INTRODUCTION

PT. XYZ decided to create simple anonymous voting system called **YukPilih**. This system will be used for asking some company decisions to their employees. Your task is to implement the backend with Laravel PHP Framework and frontend with JavaScript Framework (VueJS, AngularJS, or ReactJS). The detail description and tools that you can use will be described below.

DESCRIPTION OF PROJECT AND TASKS

API List:

Use provided ERD to make your database. Create dummy users from provided "users.csv" to users and divisions table (password is hashed using bcrypt).

These are the list of web service endpoint that requested by the company:

1. Authentication

a. Login

URL: [domain]/api/auth/login

Description: For client to generate and get login token using username and password. Username and password must be valid.

Method: POST

Request Parameter:

- Body:
 - username
 - password

Response:

- If login success:
 - Header: response status: 200
 - Body:
 - access_token (authorization token generated by the system using JSON Web Token (JWT))
 - token_type (always Bearer)
 - expires_in (set your token to be expired in 24 hours)
- If login failed (username or password do not match or empty):

Header: response status: 401

Body: error: Unauthorized

b. **Logout**

URL: [domain]/api/auth/logout

Headers:

- Authorization: Bearer [access_token]

Description: For server to make token invalid

Method: POST

Response:

- If logout success:
Header: response status: 200
Body:
 - message: successfully logged out
- If logout failed (token invalid):
Header: response status: 401
Body: message: Unauthorized

c. **Me**

URL: [domain]/api/auth/me

Headers:

- Authorization: Bearer [access_token]

Description: Get logged in user data

Method: POST

Response:

- If success:
Header: response status: 200
Body:
 - User data including id, division_id, username, role, created_at, updated_at
- If failed (token invalid):
Header: response status: 401
Body: message: Unauthorized

d. **Reset Password**

URL: [domain]/api/auth/reset_password

Headers:

- Authorization: Bearer [access_token]

Description: Reset user password. Invalidate current token if reset success.

Method: POST

Request Parameter:

- Body:

- old_password
- new_password

Response:

- If success:
Header: response status: 200
Body: message: reset success, user logged out
- If failed (old password did not match):
Header: response status: 422
Body: message: old password did not match
- If failed (token invalid):
Header: response status: 401
Body: message: Unauthorized

2. Poll

a. Create new poll

URL: [domain]/api/poll

Headers:

- Authorization: Bearer [access_token]

Description: Create new poll. Admin only.

Method: POST

Request Parameter:

- Body:
 - title: String, required.
 - description: String, required.
 - deadline: Datetime, required.
 - choices: array of string, choices must be unique. Validate at least two items given.

Response:

- If success:
Header: response status: 200
Body: JSON of the new poll data
- If unauthorized user access it (only admin can access this endpoint):
Header: response status: 401
Body: message: Unauthorized

b. Get all poll

URL: [domain]/api/poll

Headers:

- Authorization: Bearer [access_token]

Description: Get all poll data. Result only shown to admin or if user already voted for that poll or if that post already exceeds the deadline. Otherwise, result will be returned as null.

Method: GET

Response:

- If success:

Header: response status: 200

Body: array of JSON of the polls data, including id, title, description, deadline, created_by (creator user id), created_at, creator (creator username), result, and choices.

Notes: result will be an array consists of id (choice_id), choice, and point. Check detailed formula in the end of this document to learn how the point calculated.

- If using invalid token:

Header: response status: 401

Body: message: Unauthorized

c. **Get a poll**

URL: [domain]/api/poll/[poll_id]

Headers:

- Authorization: Bearer [access_token]

Description: Get one poll data by given poll_id. Result only shown to admin or if user already voted for that poll or if that post already exceeds the deadline. Otherwise, result will be returned as null.

Method: GET

Response:

- If success:

Header: response status: 200

Body: JSON of the polls data, including id, title, description, deadline, created_by (creator user id), created_at, creator (creator username), result, and choices.

Notes: result will be an array consists of id (choice_id), choice, and point. Check detailed formula in the end of this document to learn how the point calculated.

- If using invalid token:

Header: response status: 401

Body: message: Unauthorized

d. **Vote**

URL: [domain]/api/poll/[poll_id]/vote/[choice_id]

Headers:

- Authorization: Bearer [access_token]

Description: Logged user give a vote for given choice_id to the given poll_id. Admin cannot do a vote.

Method: POST

Response:

- If success:
Header: response status: 200
Body: message: voting success
- If user already voted for this poll:
Header: response status: 422
Body: message: already voted
- If poll already deadline:
Header: response status: 422
Body: message: voting deadline
- If user try to use invalid choice_id or poll_id:
Header: response status: 422
Body: message: invalid choice
- If unauthorized user access it (only user can access this endpoint):
Header: response status: 401
Body: message: Unauthorized

e. **Delete a poll**

URL: [domain]/api/poll/[poll_id]

Headers:

- Authorization: Bearer [access_token]

Description: Admin delete poll based on given poll_id.

Method: DELETE

Response:

- If success:
Header: response status: 200
- If unauthorized user access it (only admin can access this endpoint):
Header: response status: 401
Body: message: Unauthorized

The complete **YukPilih** system should cover the following requirement:

Menu	Detail
Login	<ul style="list-style-type: none"> ● The system's role can be an Admin or a User ● The user (admin or user) can login (and logout) into the system on the start page of the application ● After login, if admin or user still using default password, frontend will be redirected to change password page. Else, redirect to poll page. ● After login, admin will have floating action button (FAB) to create a new poll ● After login, user can view all schedule <p>Note: the system is not providing register menu, all user data registration is managed by administrator on the database directly</p>
Change Password	<ul style="list-style-type: none"> ● After login, if admin or user still using default password, frontend will be redirected to change password page ● On this page, user will be asked to input old password, new password, and confirm new password. Validate new password and confirm new password must be match before hitting reset_password backend endpoint. ● After success resetting password, redirect to login page.
Poll	<ul style="list-style-type: none"> ● Username and logout button will be displayed on the menu. Admin or user can use the button for logging out. ● Admin will have floating action button (FAB) to create a new poll ● Admin cannot do vote. Admin always seeing the poll results. ● Admin can delete polls. ● User can do vote only once per poll. ● User only seeing poll results only if the poll already deadline or user already voted for that poll. ● Polls sorted by created_at descending. ● Poll results will be shown as percentage with following formula: Choice's percentage = choice's point / [total of all points] x 100% ● Point will be calculated with following standards: <ul style="list-style-type: none"> ○ Regardless number of division members, a division will only have 1 point for each poll. ○ All voting will be calculated internally on each division. ○ After calculated internally, the winner from each division will be used for the whole poll. ○ If there is more than one winner on a division, the division's point will be split equally.
Create Poll Dialog	<ul style="list-style-type: none"> ● If admin click the FAB button, this dialog will be shown. ● Admin asked to fill title (string), description (string, can be multiline), deadline (use date time picker, default value is current time), and choices. ● Choices have special behaviours: <ul style="list-style-type: none"> ○ Initially, there is only one input field. ○ After admin typing down on the input field, a new blank input field will be appended in the bottom of the form. ○ Admin can delete input fields, but make sure admin always have at least one blank input field in the bottom of the form. ● There is a cancel button to close this dialog without creating poll.

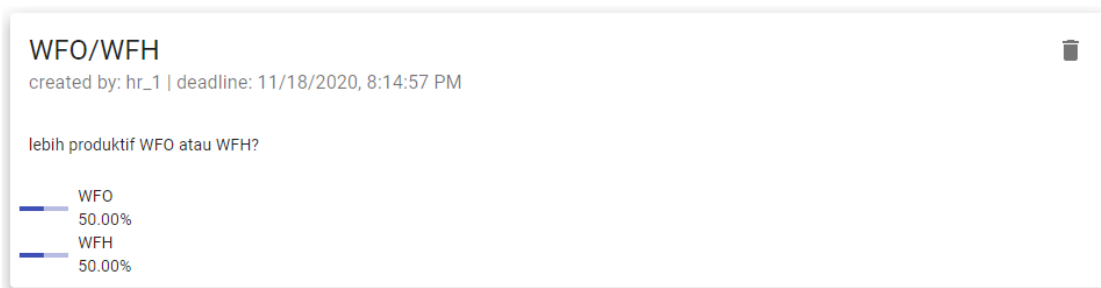
YUK PILIH POINT CALCULATION SIMULATION

Simulation 1:

This poll having 2 choices (WFO or WFH).

- Two users from payment division already vote. All of them choosing WFO. WFO win on this division, one point for WFO.
- One user from procurement division choosing WFO. WFO win on this division, one point for WFO.
- Six users from IT division already vote. 1 choosing WFO, 5 choosing WFH. WFH win on this division, one point for WFH.
- Three users from finance division already vote. All of them choosing WFH. WFH win on this division, one point for WFH.

Result: WFO 2 points. WFH 2 points. 50% percentage for each option will be displayed on the frontend.



Simulation 2:

This poll having 3 choices ('senin', 'rabu', or 'jumat').

- All users from payment division already vote. All of them choosing 'rabu'. 'rabu' win on this division, one point for 'rabu'.
- Three users from procurement division voted. 1 choosing 'senin', 1 choosing 'rabu', 1 choosing 'jumat'. We have 3 winners on this division. 1/3 point will be given for each option.
- Five users from IT division voted. 2 choosing 'senin', 2 choosing 'rabu', 1 choosing 'jumat'. We have 2 winners on this division. 1/2 point will be given for 'senin' and 'rabu'.
- No user from finance division voted. No point given from this division yet.

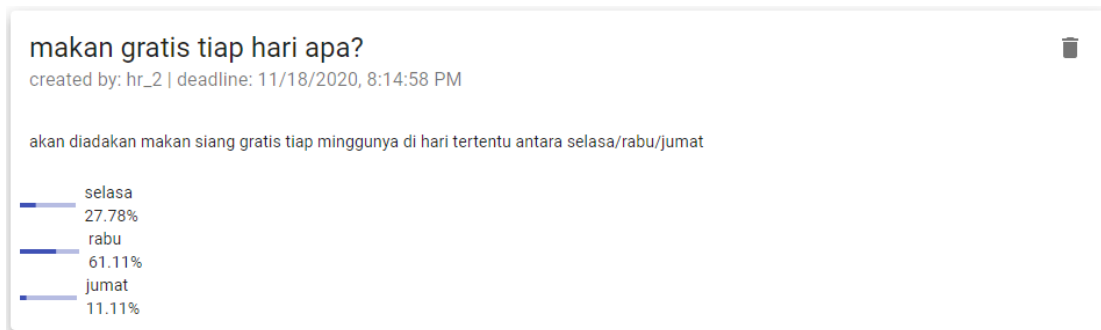
Result: 'senin' 0.8333 points. 'rabu' 1.8333 points. 'jumat' 0.3333 points.

Percentage calculation:

$$\text{'senin'} = 0.8333 / (0.8333 + 1.8333 + 0.3333) * 100\% = 27.78\%$$

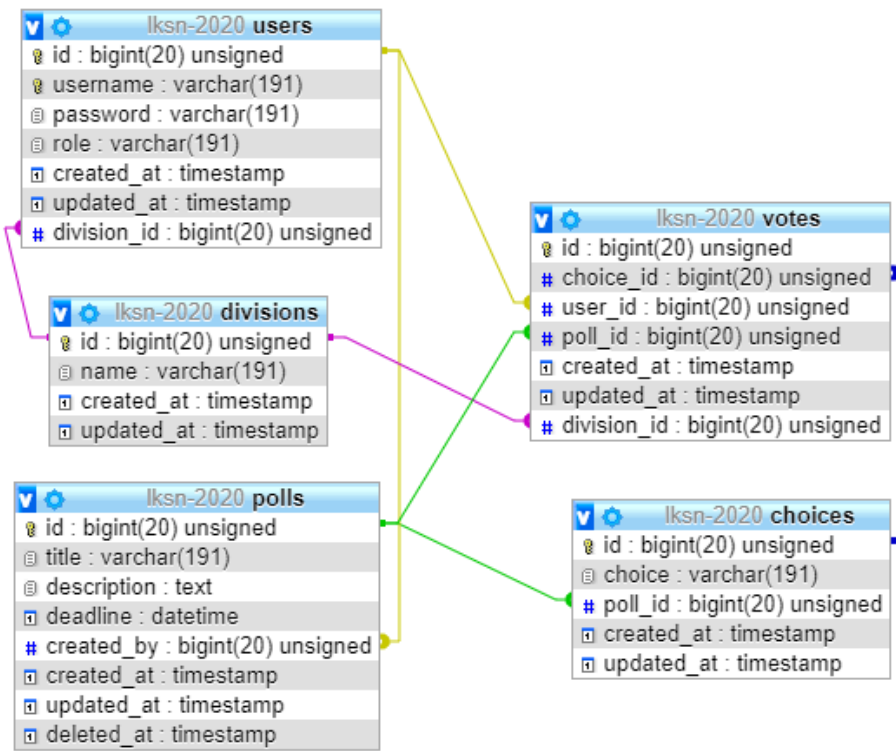
$$\text{'rabu'} = 1.8333 / (0.8333 + 1.8333 + 0.3333) * 100\% = 61.11\%$$

$$\text{'jumat'} = 0.3333 / (0.8333 + 1.8333 + 0.3333) * 100\% = 11.11\%$$



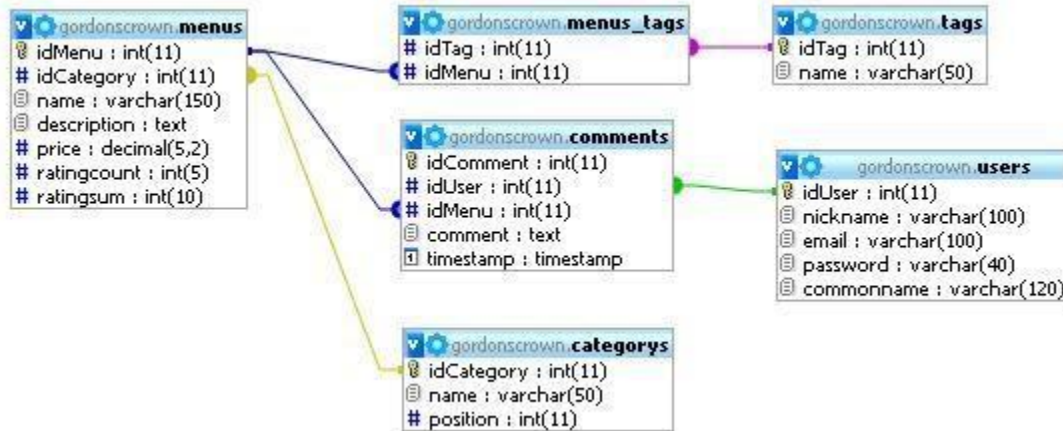
ERD

You can use and improve ERD below:



INSTRUCTIONS TO THE COMPETITOR

- Save your files in your root directory on the server called "**module_server**"
- **CREATE/GENERATE A DB-DIAGRAM NAMED "DB-DIAGRAM.XXX" (XXX IS THE EXTENSION/TYPE OF THE FILE EG. PDF OR JPG) AND PUT IT INTO THE DIRECTORY MENTIONED ABOVE. EXAMPLE:**



For this module, you must use one of the three available frameworks provided. Applications developed without use of any of these frameworks will not be considered. You should take advantage of the framework as much as possible.

CLIENT MODULE

CONTENTS

THIS MODULE HAS THE FOLLOWING FILES:

1. MODULE_CLIENT_SIDE.doc
2. MODULE_CLIENT_SIDE_MEDIA.zip

INTRODUCTION

In recent years internet has become our basic daily source and needs, enabling the dissemination of information in an inexhaustible content and interaction. Games uses has gained a prominent role in nowadays computer usage, allowing people to get access to entertainment from any places.

You are asked to develop game called **Phytons** using HTML and CSS and develop client-side programming using JavaScript and its open source libraries. Some media files are available to you in a zip file. You can create more media and modify anything in the media if you want.

Your game needs to be developed in a tablet resolution (960x600 pixels). In bigger resolution, the game must be centred in the screen both horizontally and vertically.

DESCRIPTION OF PROJECTS AND TASKS

This is a module of 4 hours. Your first 2 hours must be used to create the design of the game in three PNG images and the initial layout using HTML/CSS. Your layout should follow the design that you created. The final 2 hours you will create the functionality of game using JavaScript that allows the game to work correctly in different web browsers.

Phytons game boards are describe below:

1. Game Title
2. Board are 960 x 600 with 48x30 grids
3. Phytons which has 6 in length
4. Player score
5. Total time
6. Rewind button

DESIGN AND INITIAL LAYOUT

1. **Deliver at least 3 PNG image files that present:**

- a. Game instruction: The first screen of the game presents the instructions to the player, a text field for player's name, and the Play buttons for starting the game. The instructions for the game are included in the media files. The "Play" button should be disabled if the text field is empty.
 - b. Game board: It must present all elements described above in the game screen.
 - c. Game icon which represents the game immediately.
2. **Develop the initial markup (HTML + CSS) of your game application.** Overall screen must be within 1024x1024 pixels.
 3. **Buttons must have active and hover effects.** The buttons background in hover state must be #5d96d4 and the active state must use #0069d9.
 4. **The HTML and CSS code must be valid in the W3C standards for HTML5 and CSS3 rules in accordance with the WCAG and standard ARIA (Accessible Rich Internet Applications Suite)**

GAME FUNCTIONALITIES

1. **Show game icon** on the top of the screen.
2. **Show game instruction** in the centre after page are loaded.
3. **Show text field for player** to input their name.
4. **Player can start game** after click "Play" button at the bottom of game instruction.
5. **The "Play" button should be disabled** if the text field is empty.
6. **Phyton's starting point is in the middle of board** and moving east (from left to right) automatically 4 grid per second. The phyton moves only north, south, east, or west.
7. **Phyton cannot** move directly into opposite direction.
8. **To move phyton**, player can use arrow keys or "WASD" keys with the following:
 - a. North with up arrow or 'w'
 - b. South with down arrow or 's'
 - c. East with right arrow or 'd'
 - d. West with left arrow or 'a'
9. **Phyton should eat food pellet** to increase its length which also increase player's score.
10. **Pellet appear at random location inside board every 3 seconds** and will disappear after 5 seconds. Number of pellets in board is minimum 3 and maximum 5 at the same time.
11. **Pellets cannot appear** on python's location.
12. **Player score are based on phyton's length.** So, the initial score would be 6.
13. **There's a timer at the top of board** whose shows elapsed time and player score.
14. **If phyton head hit board edge**, then its head should appear on the opposite edge.
15. **Player can rewind the phyton game for 5 seconds.** Rewind features allows player to rewind the game anytime they want.
16. **Rewind button are located at the bottom of game board**, or player can activate it using "space" button.
17. **After the rewind feature activated**, there will be a seeker in the form of slider which player can moves below the game board, the game will rewind accordingly. This seeker can be moved 5 ticks, each tick for each second.
18. **Player can cancel the rewind feature** using "Cancel Rewind" button.

19. **Phyton dies if it hit its own tail.** Game should notify player it is over and shows Highscore. If the current game score is higher, then it should be the highscore. Highscore should be saved.
20. **Your game should work without JavaScript errors** or messages shown in the browser console.
21. **Maintain your HTML/CSS and JavaScript code organized and clean to facilitate future maintenance.** Use correct indentation and comments. Use meaningful variable names and document your code as much as possible so another developer would be able to modify your work in the future.
22. The game needs to work correctly in **Google Chrome**.

INSTRUCTION FOR COMPETITORS

1. The media files are available in the ZIP file. You can modify the supplied files and create new media files to ensure the correct functionality and improve the application.
2. Save your design files in a folder called "**module_client_side/design**".
3. You should create additional images for each of the requested resolutions to highlight hidden elements, animations, interactions, or any additional information that will assist in the presentation of the game design.
4. Additional file names
 - a. Instructions: province_instructions_2.png, XX_instructions_3.png
 - b. Game board: province_game_board_2.png, XX_game_board_3.png
5. Save any image source files to a folder named "**source**" inside the "**module_client_side/design**" folder. The source files are the files that contain the design layers, development files, i.e. xd, psd, ai, or svg.
6. Save the working game to the directory on the server named "**module_client_side**". Be sure that your main file is called index.html.
7. You are responsible for the time management in your development. If you finalize some tasks you can continue to other tasks.

Example

These following images are for example purpose only. You may design your own game layout.



Image 1 Python example



Image 2 Python Rewind Feature

