



# **Test Project**

## **Web Technologies**

2024 SELEKDA ONLINE

# Introduction

The Indonesian WebTech team needs a website to provide information about technological developments to all users in the world. In addition to providing information on technological developments, we also provide free applications and games that can be played with leaderboards, so users can measure their abilities against other players.

The website needed to provide login and register features for the user. The target audience for this website is teens and young adults who are interested in playing games created using canvas in HTML5.

They need a working website that will be stored into the hosting so that it can be easily accessed.

## Description of Project and Module

Your task as a website developer is to create a design, front end and back end for the website and make it work.

You have been provided with some images that will help you to create the website, apps and the games. You are not required to use all of the images; even you can add more text or images if necessary. You need to make a good website and working games, because the website will be accessed by users from around the country. The use of text and image will only be judged on design, not the meaning of the text.

### 1. Design

In the design part, you should design about at least 5 pages of a website. The new design should be developed to work on the defined devices and resolution: Desktop (1440 x 900 px), Tablet (768 x 1024px) and Smartphone (320 x 480px). Just remember, you can add your own creativity to your design. Don't be afraid to make something different rather than use a normal design or mainstream design.

And here are some description about design task:

1. Create a Logo for **“WebTech Indonesia”** that will be used for the organisation's logo.
  - a. The logo must be used as part of your design within the website concept
  - b. Your logo must be consist of combination between shape and text **“WebtechID”**
2. Design 5 pages, consist:
  - a. **Landing Page**
    - i. Your logo
    - ii. Interactive banner section
    - iii. Images and video gallery thumbnail section
    - iv. Short text section about the website
    - v. Fixed interactive menu on the top of website
    - vi. News
    - vii. Services
    - viii. Portfolio
    - ix. Testimonials
    - x. Searching section to search for news

- xi. Social media icon (at least three social media icon)
- xii. Acknowledgment or copyright **and** the text “Designed by : <You>” at the last section in the footer

**b. Blog**

- i. Your Logo
- ii. Blog List (image, title, description, author, date, total view and comments)
- iii. List Categories,
- iv. Popular Blog Post
- v. Tag Clouds

**c. Blog Detail**

- i. Your Logo
- ii. Blog details (image, title, description, author, date, total view, total comments)
- iii. Comment List (user image, name, date, comment)
- iv. Comment Form (name, email, subject, website, comment, captcha, submit button)
- v. Popular Tags

**d. Portfolio**

- i. Your Logo
- ii. Portfolio Categories filter
- iii. Portfolio List (image, title, description, author, date, total view and comments)
- iv. Popup Modal detail project

**e. Admin Dashboard**

- i. Your Logo
- ii. Navigation Bar
- iii. Left Sidebar
- iv. 3 Chart Statistic
- v. Search
- vi. User Info

3. You should place your work into 1 folder, which called **“design”**, you should only collect the photoshop files (\*.psd) and screenshot files (\*.jpg, \*.png) for each design. The naming files should be like “Computer\_Landing.psd”, “Computer\_Landing.jpg”, “Computer\_Blog.psd”, etc.

## 2. Layout

In this part you should slice your own design into the html. In this part, you are **not allowed to use any CSS framework**.

The page should be a responsive website similar to the design you created before. You must add some animation or interactive interaction in your own page (the banner, the transition, hover animation, etc.).

You should collect your job into 1 folder, which called **“layout”**, you should only collect the html files (\*.html), javascript files (\*.js) and css (\*.css) in the proper folder inside of the **“layout”** folder.

### Provision:

1. The website code must validate to HTML5
2. Your CSS code must validate CSS level 3
3. No error and warning found in js files
4. Place a comment in html, js and css files
5. You should collect the **fonts** if you use a new font, in the folder **“fonts”**
6. Some elements are only visible in wide screen layout, and the others can only appear in the narrow screen. You decide by your own decision, which one must be shown in the wide screen or which one must be hide
7. Your website will be marked in Google Chrome

## 3. Web Apps

In this part you are asked to develop an illustration web app called **“DSGN”** using HTML and CSS with JavaScript. Media files are available as a compressed folder. Your apps need to be developed in a 1280 x 600 pixels resolution and should be centred both horizontally and vertically.

**DSGN** workspace screen should have these features below:

1. Work area
2. Toolbar with tools
3. Properties bar
4. Layer bar
5. Logo of DSGN
6. Undo and redo button with counter for each
7. Zoom in and out button

### Design and Initial layout:

1. **Develop the initial markup (HTML + CSS) of your game application.** Overall screen must be within 1280 x 600 pixels and dead centred on the screen.
2. **The design should be delivered in dark mode colour.** You are free to choose dark colours as long as you have user convenience.
3. **You are free to beautify** the app screen as long as it meets the requirements.
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)

### Apps functionalities:

1. **Show apps welcome screen which** have a logo and app name. Welcome screen shows in the centre and should fade-out in 1 second and then a new file window will appear.
2. **User input image size and file name** and press the “create new” button on bottom right corner of new file window.
3. **Workspace will load** after “create new” is pressed. Work area is filled with white as default and named “Background” Layer.
4. **Toolbar are shown** on the left side with all tools shown.
5. **Brush tool** allows user to paint on selected layer, much like a real paintbrush and has properties of:
  - a. Brush size with slider from 1 to 100
  - b. Brush shape to choose round or square.
  - c. Brush opacity which states 0, 50, or 100 in percent
6. **Eraser Tools to delete the unwanted pixels.** Eraser should work on selected layers or all layers depend on properties selected. The Eraser tool has the same properties as the Brush tool.
7. **Move Tools helps users position** selected content or layers when customising their work simply by dragging after selecting move tools.
8. **Move Tools also have properties of** rotate clockwise, counter clockwise, flip horizontal, and flip vertical.
9. **Shape Tools** to create shape with selection of circle, square, rectangle, and line.
10. **Colour Picker Tools to pick colour** from the work area. This tool should take samples on all layers, not just on selected.
11. **Paint Bucket tools to fill adjacent pixels** that are similar in colour value to pixels you select. It fills these pixels with the selected colour.
12. **Text tools allow users to add text** to their working area. When the user selects text tools and then clicks on the chosen place on the working area, DSGN will create a text field for text on a new layer.
13. **Text tools have properties** typeface, size, and style as follow:
  - a. Typefaces show in the dropdown.
  - b. Size from 8 to 200 pt.
  - c. Style of bold, italic, or underline.
14. **To apply style,** users can customise the text properties before typing text.
15. **While typing, there will be a floating button** on top of the text field for the finish edit button represented with white check icon on a green circle.
16. **After finishing typing, the new layer created by text tools** will be renamed based on the first 10 characters of the text typed.
17. **Clone Stamp tools** to copy pixels from one area and apply them to another. Think about it as “copying and pasting,” but users are “pasting” the content by painting.
  - a. **User select the part** of they want to clone by alt + click
  - b. This action will **record a round area** of 50px across
  - c. Users can **stamp the cloned area** on any selected layer.
18. **DSGN also has a layer feature** which is a stacked, transparent sheet of glass on which users can paint images. Users can see through the transparent areas of a layer to the layers below. Users can work on each layer independently, experimenting to create the effect you want.

19. **Users can hide a specific layer** with a hide button on the left side of the layer name represented as an eye icon.
20. **Layer stack shows** on the right side of the screen as a list.
21. **Layers can be customised** as well by adjusting opacity, brightness, saturation, and greyscale. Users can select a layer, and the layer option pane on top of the colour picker area will be activated. Users can adjust by selecting the option and then slide using the same slider.
22. **To cancel layer option selection**, simply by clicking the activated option again.
23. **Selected Layer can be duplicated** by clicking the duplicate button on the layer option pane.
24. **Users can rearrange the layer** by dragging the selected layer to its new place.
25. **While dragging, tilt the selected layer** while dragged and show movement animation of the adjacent layer.
26. **Undo / Redo functionality** could be accessed on screen with counters for each action. Users can undo 5 times and redo as many actions as they have done. Undo can also perform using shortcuts of ctrl + z and redo using ctrl + y.
27. **DSGN also has a Selection Tool** which isolates part of an image so the user can work on that area without affecting the rest of the image. In the toolbar, select the Rectangular Marquee tool. Drag a rectangular selection onto the image.
28. **While selected, there will be a floating button** which has a copy function.
29. **The selected part of the layer can be deleted** using the Delete key.
30. **If there's an object on the clipboard**, the user can click the paste button on the toolbar to paste the copied object on a new layer. Otherwise the paste button will be inactive.
31. **Users can Import an image** from a file to DSGN apps using the import button on the toolbar and it will appear as a new layer on top of the selected layer.
32. **Export file to jpg** using the export button on the toolbar.
33. **Colour select pane on top of layer stack** in which user can pick colour using RGB slider and it will preview the composed colour.
34. **Zoom in and zoom out button** to zoom in and out working area. Users can also see the percentage of zoom applied to the work area.
35. **DSGN needs to work correctly** in Google Chrome.

Please include comments in your JavaScript files to indicate the functionalities of the code. You should collect the **html files** (\*.html), **javascript files** (\*.js), and **media files** in a folder called "apps".

This following image is for example purpose only. You may make changes or design your own layout.

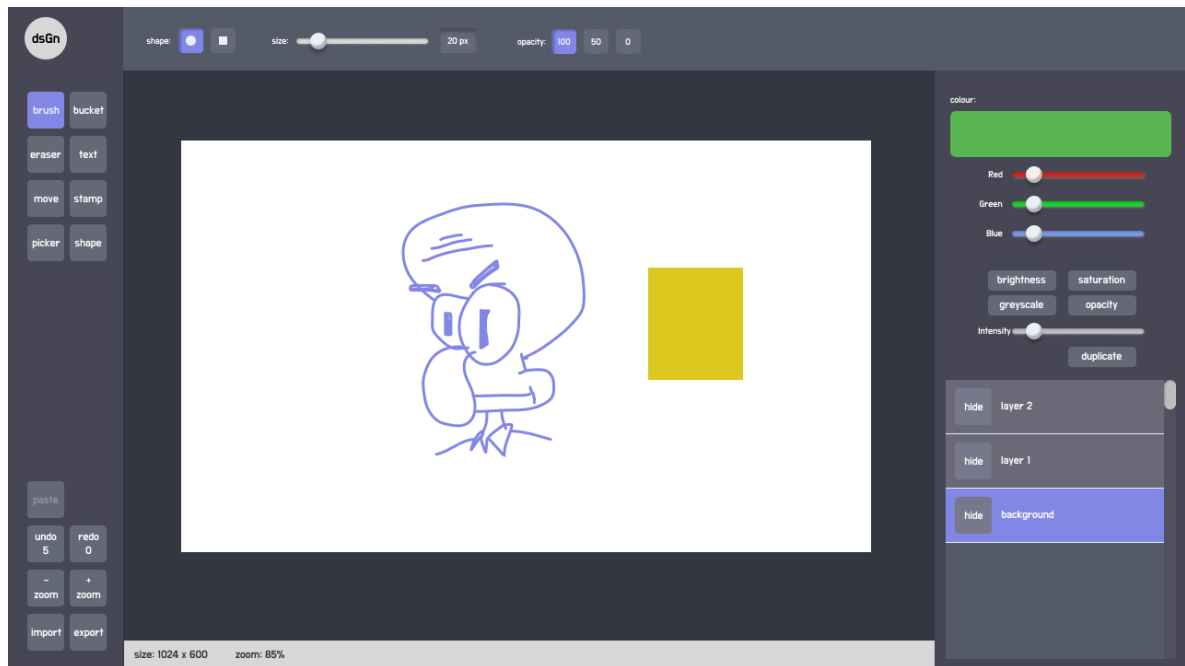


Image 1 Example Web Apps

#### 4. Game

You are asked to develop a game called World Head Football using HTML and CSS and develop client-side programming using JavaScript. 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 (1000 x 600 pixels). In bigger resolution, the game must be centred in the screen both horizontally and vertically.

**World Head Football game screen should have meet these requirements below:**

1. Player Name
2. Gameboard
3. Player Character
4. Country Flag
5. Total Score
6. Timer
7. Match History

**Design and Initial layout:**

- a. **Develop the initial markup (HTML + CSS) of your game application.** Overall screen must be within 1000 x 600 pixels and centred on the screen.
- a. **The design should be delivered in dark mode colour.** You are free to choose dark colour as long as it has the user convenience.
- b. **You are free to decorate** the game screen design as long as it meets the requirements.
- c. **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 welcome** in the centre after the page is loaded.
2. **Players can go to the game** after filling the username field and click the **"Play Game"** button at the bottom of the welcome page.
3. **The "Play Game" button should be disabled** if the user did not input the username.
4. Users can choose one of three levels (easy, medium, hard).
5. Users can choose one of two balls to use.
6. Users can choose one of many countries to use.
7. Users can **randomise** one of many countries as an opponent (computer).
8. The opposing country cannot be the same as the one we chose.
9. Users can show instructions after clicking the **"Instruction"** button.
10. Users can close instructions after clicking the **"X"** button.
11. **Show countdown for three seconds in the centre of screen** after the user clicked the play button before the game started playing.
12. **When the game starts**, the timer will start with time according to level.
  - a. '30 seconds' for easy level
  - b. '20 seconds' for medium level
  - c. '15 seconds' for hard level
13. The player will be on the left and the computer on the right.
14. The ball will appear from top to bottom in **the middle of 2 characters**.
15. **The ball will bounce** when it hits the body part of the character
16. Player can use the following buttons to set the character:
  - a. "A" to move left
  - b. "W" to jump
  - c. "D" to move right
  - d. "Space" to kick
17. **Drops items** every 5 seconds.
18. **The items that will drop** are as follows:
  - a. Increase Ball box to increase ball size
  - b. Decrease Ball box to decrease ball size
  - c. Diamond ice to freeze ball in 3 seconds
19. **The item disappears** after being hit by the ball
20. **Walking animation** when the character is moving.
21. **The opponent (computer) can move as needed** and score goals.
22. **The score will be increased** if the player can score a goal against the opponent.
23. After a goal, **the ball will disappear** and reappear from top to bottom in the **middle of 2 characters**.
24. **Players can pause** the game.
25. **Press Esc to open the pause popup**. The game should be in a paused state when opening the popup.
26. Press **Esc** again to **continue** or click the **continue** button.
27. **Game Over** if the timer time is up.
28. But if the time is over but the score is still the same, then one of the characters must score again, then the game is over
29. **Show popup after game over** to display the player username, playing country, score, save match history button and restart button.
30. Match history should be saved in the database after the player clicks the **Save Match History** button.
31. Players can see the match history after clicking the Match History Button.



32. Match history can be sorted by score and last matches.
33. The game needs to work correctly in Google Chrome.

Please include comments in your JavaScript files to indicate the functionalities of the code. You should collect the **html files** (\*.html), **javascript files** (\*.js), and **media files** in a folder called “game”.



**Image 2** Example Gameboard

## 5. Server Side

In this part you should only use the framework **laravel**. For the database you can use **MySQL** or **mariadb** as the database. You should create a RESTful API with several features listed below using the PHP framework. Please note that you need to follow [Google JSON Style Guide](#) in your JSON response.

**There are some function that should be able to be used:**

- 1) Web token to authorised user access to the website API (using **sanctum**)
- 2) API for several feature listed below:
  - a) User API
    - i) User login (Admin and User)
    - ii) User registration, with fields:
      - (1) Name
      - (2) Username
      - (3) Email
      - (4) Password
      - (5) Date of Birth
      - (6) Phone Number
      - (7) Profile Picture
      - (8) Date
    - iii) User profile update
    - iv) Get user profile
    - v) Blog Comments (View, Insert, Update, Delete), with fields:
      - (1) Name
      - (2) Email
      - (3) Subject

- (4) Website
  - (5) Comment
  - (6) Captcha
  - (7) Date
- b) Captcha API
  - i) Get captcha
  - ii) Validate captcha
- c) Data Management API (Admin Only)
  - i) Banner (View, Insert, Update, Delete), with fields:
    - (1) Banner Title
    - (2) Banner Image
    - (3) Description
    - (4) Status (active, inactive)
    - (5) Date
  - ii) Blog (View, Insert, Update, Delete), with fields:
    - (1) Blog Image
    - (2) Blog Title
    - (3) Description
    - (4) Author
    - (5) Tags
    - (6) Date
  - iii) Portfolio (View, Insert, Update, Delete), with fields:
    - (1) Portfolio Title
    - (2) Portfolio Image
    - (3) Description
    - (4) Author
  - iv) User, with fields:
    - (1) Name
    - (2) Username
    - (3) Email
    - (4) Password
    - (5) Date of Birth
    - (6) Phone Number
    - (7) Profile Picture
- d) Selekdash game API
  - i) Get leaderboard
  - ii) Save score to leaderboard

You should collect the **picture of database diagram** (\*.png), **your zip project's code** (\*.zip), **database** (\*.sql), **postman collection** and **API documentation** (\*.doc) in the folder, which is called the “**server**” folder.

## 6. Implementation

In this part you should combine all of the work you created before. You need to make the front end exactly like your designed, using your sliced html which is ported to be the website front end using one of JS Framework (VueJS or ReactJS), and implement the games also you need to connect your games and front end with the RESTful API you created.

There is several things that you should make in this section:

1. Login & logout for admin & user
2. Admin has several feature listed below:

- a. Data Management
    - i. Banner (View, Insert, Update, Delete), with fields:
      - a) Banner Title
      - b) Banner Image
      - c) Description
      - d) Status (active, inactive)
      - e) Date
    - ii. Blog (View, Insert, Update, Delete), with fields:
      - a) Blog Image
      - b) Blog Title
      - c) Description
      - d) Author
      - e) Tags
      - f) Date
    - iii. Portfolio (View, Insert, Update, Delete), with fields:
      - a) Portfolio Title
      - b) Portfolio Image
      - c) Description
      - d) Author
    - iv. User, with fields:
      - a) Name
      - b) Username
      - c) Email
      - d) Password
      - e) Date of Birth
      - f) Phone Number
      - g) Profile Picture
  - b. Update Profile
3. User has several feature listed below:
    - a. Update Profile
    - b. My Leaderboard
    - c. Play Game
    - d. DSGN Illustration
  4. Save score to leaderboard
  5. Make sure that only logged in users can play the game, if the guest clicks the Play button, show only Login & Register popup modal.
  6. Everytime a user plays the game, save the final score to the leaderboard so other users can see who got the top 10 highest score.
  7. Leaderboard which displayed beside the game canvas.
  8. When a user wants to login and register, please add a simple captcha to prevent bruteforce attack. Note that you must create the captcha yourselves.

Collect the final result of your work in a folder, which is called the “**front\_end**” folder.

# Submission

Please note that all your work must be uploaded to a hosting platform for the apps. Additionally, uploading the source code is mandatory; package all the source code and required files in a single file named **selekda.zip**.

Make sure to provide your website address when submitting your work.

**Example :**

Demo : **webtech.com**

Source Code : **webtech.com/selekda.zip**

Structure folder in **selekda.zip**



There is no limitation on the code, as long as you provide clear instructions (such as a README) and ensure that everything runs smoothly. You may add additional pages or functions if necessary.

Participants are required to **commit their work every 30 minutes**. If you plan to take a break, ensure that you commit your work before and after the break. You may use reminders to make sure you make a commit.

Each feature must include a **Postman collection and clear documentation**. Any API must be documented with Postman; APIs without documentation will not be evaluated. Your project documentation should be comprehensive, covering how to access the project, user credentials, and other necessary details. Projects that cannot be accessed will not be evaluated. This documentation must be saved inside the server folder.

All work must be **uploaded to a hosting platform**. Scoring will be based on the demo URL you provide. If the provided URL is not functional, the project will automatically receive zero points.

We do not specifically recommend any hosting service. You are free to use any hosting platform of your choice.

–Good luck!--