Republic of Rwanda
Ministry of Education

RTB | RWANDA TVET BOARD

VUE.JS FRAMEWORK

SWDVF301

DEVELOP SIMPLE GAME IN VUE

Competence

RQF Level: 3

Learning Hours 150

Credits: 15

Sector: ICT & MULTIMEDIA

Trade: SOFTWARE DEVELOPMENT

Module Type: Specific

Curriculum: SWDGVF3001- TVET Certificate 3 in software development

<table>
<tr><td>Purpose statement</td><td colspan="6">Upon completion of this specific module, trainees will be able to: Set up environment,Apply vue framework and develop simple game in vue js.</td></tr>
<tr><td rowspan="6">Delivery modality</td><td colspan="2">Training delivery</td><td>100%</td><td colspan="2">Assessment</td><td>Total 100%</td></tr>
<tr><td colspan="2">Theoretical content</td><td>30%</td><td rowspan="4">Formative assessment</td><td>30%</td><td rowspan="4">50%</td></tr>
<tr><td colspan="2">Practical work:</td><td rowspan="3">70%</td><td rowspan="3">70%</td></tr>
<tr><td>Group project and presentation</td><td>20%</td></tr>
<tr><td>Individual project /Work</td><td>50%</td></tr>
<tr><td colspan="2"></td><td colspan="3">Summative Assessment</td><td>50%</td></tr>
</table>

Elements of Competency and Performance Criteria

Elements of competency	Performance criteria
1: Set Up Environment	1.1.Tools are appropriately configured according to the Vue framework
	1.2. Packages are properly installed based on Vue framework
	1.3. Bootstrap is properly setup based on styling needs
	1.4. Folders are precisely created based on the best practices of the Vue framework
2: Apply Vue Framework	2.1. Routes are correctly created in line with project pages
	2.2. Reusable components are correctly developed in accordance with most reusable HTML elements
	2.3 Form data are properly handled based on user story
	2.4. Form data are correctly validated based on user stories

	2.5. Features are correctly developed in accordance with user stories
	2.6. API requests are correctly made in accordance with user stories
	2.7. State of data is appropriately managed in accordance with user stories
3: Plan game	3.1. Storyline is methodically created based on game objectives
	3.2. Game control is properly determined based on the storyline
	3.3. Game interface is properly determined based on storyline
4: Develop Game	4.1. Game characters are properly designed based on the storyline
	4.2. Game environment is properly designed based on storyline
	4.3. Game is correctly developed based on Vue framework
	4.4. Game is correctly deployed based on static hosting platform steps

Course content	
Learning outcomes	At the end of the module the learner will be able to: - Set up environment - Apply Vue framework - Plan game - Develop game
Learning outcome 1: Set Up Environment	Learning hours: 30
Indicative content	

- Description of key concepts
  - ✓ CLI
  - ✓ IDE
  - ✓ Frontend
  - ✓ Backend
  - ✓ Single Page Application
  - ✓ NodeJs & NPM
  - ✓ Dependencies
  - ✓ Environments
    - Development
    - Testing
    - Production
  - ✓ Introduce VueJS Framework
- Vue project installation

✓ Install NodeJs

- Verify NodeJs installation
- Configure NPM
- Test javascript file using Nodejs
- Install Vue CLI with npm
- Initiate Vue Project using terminal
- Run Vue project

- **Description of Vue project folder & files**
  - ✓ Node _modules
  - ✓ Public folder
  - ✓ src
  - ✓ Asset
  - ✓ Components
  - ✓ helloWorld.vue
  - ✓ app.vue
  - ✓ main.js
  - ✓ App.vue
  - ✓ Package.json
  - ✓ Vue.config.js
  - ✓ .git ignore
  - ✓ babel.config.js

Resources required for the learning outcome	
Equipment	Computer Projector
Materials	notebook internet pencil pen
Tools	Text Editor (vscode)

	nodejs vue framework
Facilitation techniques	Demonstration and simulation Individual and group work Practical exercise Individualized Group discussion
Formative assessment methods /(CAT)	Written assessment Presentation Performance assessment Product based assessment Project based assessment Etc

Learning outcome 2: Apply Vue framework	Learning hours: 40
Indicative content	
● Definition of key concepts ✓ Components ✓ Routes ✓ Vue lifecycle ✓ State management ✓ API Endpoint	

- ✓ .env file
- Create folder structure
  - ✓ Assets folder
  - ✓ Source code folder
  - ✓ Components
  - ✓ Router folder
  - ✓ Store folder
  - ✓ Views folder
  - ✓ Mixins folder
- Apply Vue component structure
  - ✓ Create View components in views folder
  - ✓ Create Reusable components in component folder
  - ✓ Apply Bootstrap to Vue components
  - ✓ Reuse components in multiple places
- Apply navigation in Vue project using router
  - ✓ Install Router package (vue-router)
  - ✓ Create javascript file in router folder
  - ✓ Define routes array in router instantiation
  - ✓ Create view components (pages)
  - ✓ Declarative navigation
  - ✓ Use nested routes

- ✓ Use parameters inside the router
- ✓ 404 Page

- Data manipulation in Vue
  - ✓ Import necessary packages & components
  - ✓ Apply Vue lifecycle methods
  - ✓ Use Vue layout components
  - ✓ Display JSON data in a table
  - ✓ Use form in Vue component

  Create form inputs

  Input binding

  Validate form inputs

  Submit form data
- API requests
  - ✓ Install axios package
  - ✓ Configure axios in API helper file
  - ✓ Use environment variable
  - ✓ Fetch all CRUD APIs and display data to component
- Manage data using state management
  - ✓ Definition of Key concepts
    - Getter
    - Action
    - Mutation
    - Dispatch

- ✓ Benefits of State management
- ✓ State managements
  - Vuex
  - Redux
  - Pinia
- ✓ Install Vue DevTool in a browser
- ✓ Install state management(Vuex)
- ✓ Configure Vuex
- ✓ Define state modules
  - State data
  - Action
  - Mutation
  - Getters
- ✓ Store and retrieve data in state management
  - Get data from state getters
  - Commit mutations
  - Dispatch actions

Resources required for the indicative content	
Equipment	Computer projector
Materials	notebook internet pencil

	pen
Tools	Text Editor (vscode) nodejs vue framework
Facilitation techniques	Demonstration and simulation Individual work Practical exercise Individualized Trainer guided Group discussion
Formative assessment methods /(CAT)	Written assessment Oral presentation Performance assessment Product based assessment Project based assessment

Learning outcome 3: Plan game	Learning hours:30
Indicative content	
● Description of key concepts ✓ Game	

- ✓ Game types
- ✓ Narrative
- ✓ Storyline
- ✓ Game controller
- ✓ Game Settings
- ✓ Game control
- ✓ Game HUD(heads-up display)
- ✓ Game characters
- ✓ Game environment
- ✓ Game interface
- ✓ Game consoles

- Description of the Game
  - ✓ Definition of Game
    - Game type
    - Game objective
    - Game target devices
    - Game dimension
    - Game perspective

- Creation of Narrative
  - ✓ Storyline
  - ✓ Sounds

- ✓ Background music
- ✓ Environment (scenery)
- ✓ Game level / reward level
- ✓ Mission: main and side

● . Game mechanics

- ✓ Key elements for defines game mechanics
  - game hud (heads-up display)
  - Steps of the game
  - Scores
  - Level
  - Speed
  - Time
  - Target Device
- ✓ Determine game mechanics

● Identification of game controls/.

- ✓ Inputs/keys
- ✓ Hand accessibility
  - Primary control: thumb and index
  - Secondary control: Middle fingers
  - Support: Ring & pinkie fingers
- ✓ Type of game controllers

● Identification of Game Interface

- ✓ Splashscreen

- ✓ Game characters
  - Define playable characters
  - Define Non-playable Characters
  - Define characters relationship
  - Characters Interactivity
  - Elements of good characters
- ✓ Game environment
  - Define Game Dimensions
  - Define Game perspective
  - Define Playing Zone / Game Boundaries
  - Define Scenes of different levels
  - Define design tools for environment
- ✓ Alert messages (success, failure, information, warning)
- ✓ Game Play Guide

Resources required for the indicative content	
Equipment	Computer projector
Materials	- notebook - internet - pencil - pen
Tools	- Illustrator - Canvas HTML tag

	- SVG HTML tag - SASS
Facilitation techniques	Lectures Demonstration and simulation Individual and group work Practical exercise Individualized Trainer guided Group discussion
Formative assessment methods /(CAT)	Written assessment Oral presentation Performance assessment Product based assessment Project based assessment

Learning outcome 4: Develop Game	Learning hours: 50	
Indicative content		
• Definition of key concepts ✓ Deployment ✓ Deployment/Hosting platforms ✓ Domain name ✓ SASS ✓ CANVAS		

✓ SVG

- Design game interface
  - ✓ Design game environment
    - Setup Html Canvas
    - Draw in canvas HTML tags using Js
    - Style Environment using SASS
  - ✓ Design environment components with SVG or Illustrator
  - ✓ Design game HUD (heads-up display)
    - Design Containers for game stats
    - Design container for character stats
    - Design container for character resources (armor, weapon, tools,...)
  - ✓ Design game characters
    - Design characters using Illustrator
    - Design characters with SVG
- Develop game functionalities
  - ✓ Develop Game Settings page/section
  - ✓ Declare and Bind variables
  - ✓ Setup animation speed
  - ✓ Listen to Events
  - ✓ Set up game conditions
  - ✓ Setup random mechanisms to create diversity in the game
  - ✓ Setup loops for repeatable actions including Non-playable character movements

- ✓ Develop SetIntervals for timed repeatable actions
- ✓ Setup incrementals for game scores and increase game difficulties
- ✓ Design and display Alert messages
- ✓ Store data in state management

- Deploy game project on Netlify
  - ✓ Create deployment account
  - ✓ Connect project with Git repository
  - ✓ Configure deployment commands
  - ✓ Create and merge PR on Github
  - ✓ Success: Test Provided Netlify Domain

Integrated/Summative assessment (For specific module)

Integrated situation

X Rwandan museum is a museum located in Musanze district, Muhoza sector, they have a campaign directed toward educating children about historical figures and their contribution to our history.

In the beginning this campaign was conducted via historians in the museum explaining the children about those historical figures, but this method was ineffective since children would get bored and stop paying attention.

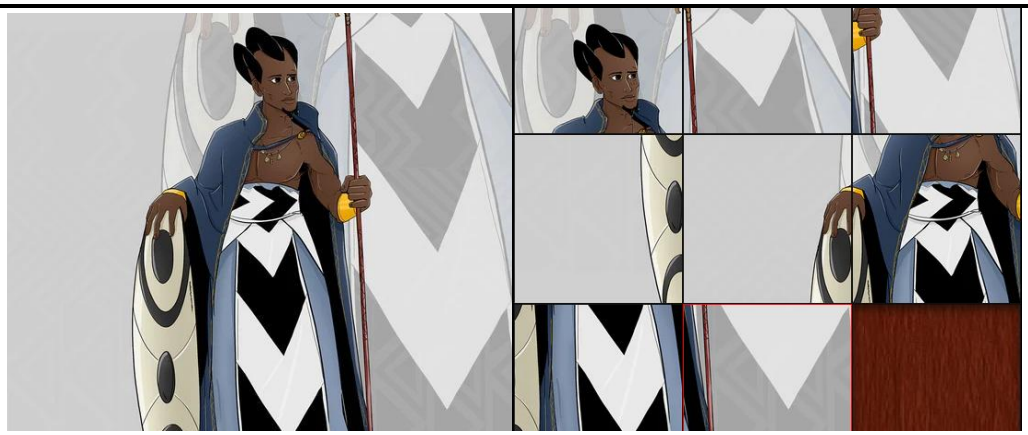
X Rwandan Museum would like to hire a game developer, to build a game where children would learn while having fun.

The game to develop will be a picture slider puzzle, where the user will get a picture with pieces arranged randomly and will have to rearrange them by clicking on the piece to move. the game must meet the following Instructions:

The puzzle will have 8 pieces and at the beginning the 9th slot will be empty
At the beginning the pieces will be arranged randomly and the sequence in which will change every you reload or try again
The user can only move pieces next to the empty slot
the pieces will be numbered depending on their respective slot they belong to
The game will container 10 picture the player can unlock by completed the puzzle
The game ends when rearranged the piece to form a clear image accor
ding to the numbering order.

when the player wins the puzzle, they will get rewarded by scores and fun fact and information about the historical figure
The pieces must be equal in size

The task must be done in 8 hours



All tools,materials and equipment are provided by the museum.

Resources

Tools	
Equipment	computer
Tools	Text Editor (vscode) nodejs vue framework
Materials/ Consumables	notebook internet pencil pen

Assessable outcomes	Assessment criteria (Based on performance criteria)	Indicator	Observation		Marks allocation
			Yes	No	
Set Up Environment **(10%)**	Tools are appropriately configured according	vscode is configured			2
		nodejs is installed			2

	to the Vue framework				
	Packages are properly installed based on Vue framework	vuejs is configured			2
		NPM is Configured			2
		sass is configured			2
Apply Vue Framework **(30%)**	Routes are correctly created in line with project pages	folder structure structure are created			2
		Router package is Installed			2
		navigation in Vue project using router is used			2
	Reusable components are correctly developed in accordance with most reusable HTML elements	Reusable components in component folder properly created			3
		Bootstrap applied on those components			3
		components are used based to multiple places			3
	Form data are correctly validated	form inputs created			3

	based on user stories	Input binding implemented			3
		The form of Inputs are Validated			3
		form data submitted			3
	API requests are correctly made in accordance with user stories	Axios package is installed			3
		environment variable is used			3
		CRUD APIs and display data to component are fetched			3
	State of data is appropriately managed in accordance with user stories	state management (Vuex) is installed			4
		state modules defined			4
		Data in state management are store and retrieve			3
Plan game (20%)	3.1. Storyline is methodically	Narrative is created			2

	created based on game objectives	Game mechanics is determined			3
	3.2. Game control is properly determined based on the storyline	Inputs/keys properly			3
		Hand accessibility is selected			3
	Game interface is properly determined based on storyline	Splash Screen is selected			3
		characters is selected			3
		Game Play Guide is created			3
Develop Game (40)	Game environment is properly designed based on storyline	Environments are created using HTML			4
		Environments are styled using SASS			4
	Game characters are properly designed based	characters are created using HTML			4

	on the storyline	characters are styled using SASS			4
		characters are interactive used vue js			4
	Game is correctly deployed based on static hosting platform steps	deployment account is created			4
		Project with Git repository is connected			4
		deployment commands are configured			4
		PR on Github is created and merged			4
		Netlify link is shared			4
Total marks					100
Percentage Weightage					100%

Minimum Passing line % (Aggregate): 70%

References:

https://www.masterclass.com/articles/how-to-design-a-video-game-character#what-makes-a-good-video-game-character

https://www.pluralsight.com/guides/getting-started-with-nodejs
https://fjolt.com/article/vue-lifecycle-hooks
https://tudip.com/blog-post/structure-a-large-scale-vue-js-application/
https://www.youtube.com/watch?v=sjYxRlwHvsM
https://www.youtube.com/watch?v=77rJ4g_aElU
https://www.youtube.com/watch?v=IxEg_Dr2utc(retrived on 26,June,2022)
https://WWW.openclassrooms.com/en/courses/5664336-create-a-web-application-with-vue-js

Glossary:

FULL FORM OF WORDS

CLI:Command-Line Interface

IDE:Integrated development environment

NPM: Node Package Manager

CRUD: Create Read Delete

SASS: Syntactically awesome style sheets

SVG: Scalable Vector Graphics

DEFINITION OF WORDS

Term 1: NodeJs : Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser, which was designed to build scalable network applications.

Term 2: NPM : npm is the package manager for the Node JavaScript platform. It puts modules in place so that node can find them, and manages dependency conflicts intelligently. It is extremely configurable to support a wide variety of use cases. Most commonly, it is used to publish, discover, install, and develop node programs.

Term 3: Vue js : is a JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS and JavaScript, and provides a declarative and component-based programming model that helps you efficiently develop user interfaces, be it simple or complex.

Term 4: Vuex : Vuex is a state management pattern + library for Vue. js applications. It serves as a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion.

Term 5: Redux: Redux is an open-source JavaScript library for managing and centralizing application state. It is most commonly used with libraries such as vue, React or Angular for building user interfaces.

Term 5: Pinia: A lightweight state management library for Vue. js.It allows you to share a state across components/pages**.**

Author's Note Page

Authoring institution

Copies available from:

Authoring institution's Adrress