



NATIONAL PRACTICAL EXAMINATION FOR TECHNICAL SECONDARY SCHOOLS (TSS)
SCHOOL YEAR 2024-2025
SUMMATIVE/INTEGRATED ASSESSMENT
SINGLE ASSESSMENT FORM

RQF Qualification Details		Assessment Center Details	
TVET SECTOR:		Place of Assessment/ Practical Examinations center:	
TRADE:			
RQF LEVEL:	5	Date of Assessment:	
QUALIFICATION TITLE:	TVET CERTIFICATE V in		
Assessed Candidates (Names and Registration Number)		Index/Reg. Number	Names

1. INTEGRATED SITUATION

Scenario

SmartPark is a company located in Rubavu District, western province of Rwanda. It provides multiple car-related services. It is struggling with inefficiencies due to its manual, paper-based system for managing parking space sales. The parking space manager records slot details such as numbers and status, and car details such as plate number, entry time, exit time, driver's name and phone number on paper. Upon a car's exit, the manager manually calculates and records the parking duration, determines the bill, and updates the slot status for new arrivals. This process is slow, prone to errors and makes it difficult to track available spaces and generate invoices efficiently. To address these challenges, SmartPark needs a web-based application that handles the parking management process. The system should allow the parking sales manager to record car entries and exits digitally, automatically calculate parking fees and update slot availability in real time.

TASK: Develop Parking Space Sales Management System (PSSMS)

As a full stack developer, you are given 7 hours to develop that web-based application by:

1. Using attributes provided below, design an Entity Relationship Diagram (ERD) that represents the relationship between their entities.
 - Identify appropriate **primary keys** and **foreign keys** based on the relationships among the entities.
 - Entities and attributes are:
 1. **ParkingSlot**(SlotNumber, SlotStatus)
 2. **Car**(PlateNumber, DriverName, PhoneNumber)
 3. **ParkingRecord**(EntryTime, ExitTime, Duration)
 4. **Payment**(AmountPaid, PaymentDate)

- ERD should be drawn before using the computer.
 - ERD should be drawn on plain paper using pencils.
 - ERD should indicate cardinalities, relationships with correct symbols
2. Creating database called **PSSMS** with ParkingSlot, Car, PakingRecord, and Payment tables as designed in ERD.
 3. Saving your work in your real names in a folder called (FirstName_LastName_National_Practical_Exam_2025).
 4. Preparing React.js Front-end application development environment by installing required modules and dependencies
 5. Preparing JavaScript runtime environment for Node.js
 6. Creating react.js components with UI features that will enable user to input his data in the tables above and display required reports.
 - Menu bar/Pages of web application should include Car, ParkingSlot, ParkingRecord, Payment, Reports and Logout options.
 - The web application should be responsive
 7. Creating react.js components with UI features that will enable user to input his data in the tables above and display required reports.
 8. Developing backend and frontend of parking sales management system.
 - Name backend project folder as **backend-project** and frontend project folder as **frontend-project**.
 - Use Tailwind CSS to implement UI design.
 - Your backend application should communicate to MySQL database/MongoDB to perform CRUD operations of using Node.js runtime environment and express.js framework.

Note: 1.Insert operation should be used on all four (4) forms (Car, ParkingSlot, ParkingRecord and Payment)

2. Delete, update, retrieve operations should only be used on


ParkingRecord form


9. Creating react.js components with UI features that will enable user to input his data in the tables above and display required reports.
10. Creating a session based login user account having username and password.
 - Password must be strong and encrypted
11. Integrating your react.js application to backend application using Axios to enable interaction of your react.js application with node.js application.
12. Generating bill containing PlateNumber, EntryTime, ExitTime, Duration, AmountPaid and PaymentDate.
 - Parking fee calculations must be hourly based.
 - Drivers should be charged 500Rwf per hour. If the parking duration is under one hour
13. Generating daily parking payment report indicating PlateNumber, EntryTime, ExitTime, Duration and AmountPaid .
14. Removing permanently your project with its related configurations after being marked.
 - Ask permission to the assessor before removing your project.

Note to the assessors:

- An assessor may promptly intervene to prevent the unnecessary use of internet and any other injury/incident that may occur during assessment
- The assessor should allow lunch time (one hour) break for candidate
- The assessor should ensure that each candidate is given time to exhibit his/her project by answering the following question:
 1. Describe your product in brief.
 2. Which difficulties have you met along the implementation of the project
 3. How did you overcome the difficulties
- The assessor should make sure that the distance between candidates prevents cheating
- The assessor should collect all ERD drawn by the candidates together with the exported project as evidences.

2. ASSESSMENT CHECKLIST

Note: Write this candidate information 	Candidate's Name	(1)			(2)			(3)			(4)		
			
			
			
	Index number												
	Starting Time												
	Ending Time												
Assessment criteria	Indicators and Elements of indicators/competencies	Yes or No	Candidate Score	Max score	Yes or No	Candidate Score	Max score	Yes or No	Candidate Score	Max score	Yes or No	Candidate Score	Max score
1. Preliminary Activities performance (15%)	Indicator 1.1: ERD is properly designed based on system requirements												
	✓ Car entity is drawn			1			1			1			1
	✓ ParkingSlot entity is drawn			1			1			1			1
	✓ ParkingRecord entity is drawn			1			1			1			1
	✓ Payment entity is drawn			1			1			1			1
	✓ Entity Symbol is used			1			1			1			1
	✓ Relationship symbol is used			1			1			1			1
	✓ Link Symbol is used			1			1			1			1
	✓ Primary Key rule is respected			1			1			1			1
	✓ Foreign Key rule is respected			1			1			1			1
	✓ Cardinalities are indicated			1			1			1			1
	✓ Car and ParkingRecord relationship is indicated			1			1			1			1
	✓ ParkingSlot and ParkingRecord relationship is indicated			1			1			1			1
	✓ ParkingRecord and Payment relationship is indicated			1			1			1			1
✓ ParkingSlot primary key is indicated			1			1			1			1	

Note: Write this candidate information 	Candidate's Name	(1)			(2)			(3)			(4)		
			
			
			
	Index number												
	Starting Time												
	Ending Time												
	✓ Car primary key is indicated			1			1			1			1
	✓ Payment primary key is indicated			1			1			1			1
	✓ ParkingRecord primary key is indicated			1			1			1			1
✓ Car foreign key is indicated in ParkingRecord			1			1			1			1	
✓ ParkingRecord foreign key is indicated in payment			1			1			1			1	
✓ ParkingSlot foreign key is indicated in ParkingRecord			1			1			1			1	
Sub-total/20		/20		/20		/20			
Percentage (15%)			
2. Process and fulfillment of the task (50%)	Indicator 2.1: Development environment is properly arranged based on coding architecture methodology												
	✓ Project folder is named as FirstName_LastName_National_Practical_Exam_2025			1			1			1			1
	✓ Node.js project is created			2			2			2			2
	✓ Express.js is installed			1			1			1			1
	✓ Cors is installed			1			1			1			1
	✓ Nodemon is installed			1			1			1			1
	✓ MySQL/MongoDB is installed in Node.js			1			1			1			1
	Indicator 2.2: React.JS environment is properly prepared based on application requirements												
	✓ React project is created			2			2			2			2
	✓ React-router-dom is installed			1			1			1			1

Note: Write this candidate information



Candidate's Name	(1)			(2)			(3)			(4)		
.....		
.....		
.....		
Index number												
Starting Time												
Ending Time												
✓ Axios is installed			1			1			1			1
Indicator 2.3: Data definition language commands are effectively applied based on database schema												
✓ PSSMS database is created			1			1			1			1
✓ User table is created			1			1			1			1
✓ Car table is created			1			1			1			1
✓ Payment table is created			1			1			1			1
✓ ParkingSlot table is created			1			1			1			1
✓ ParkingRecord table is created			1			1			1			1
✓ Primary key is applied in User table			1			1			1			1
✓ Primary key is applied in Car table			1			1			1			1
✓ Primary key is applied in ParkingSlot table			1			1			1			1
✓ Primary key is applied in Payment table			1			1			1			1
✓ Primary key is applied in ParkingRecord table			1			1			1			1
✓ ParkinRecord Foreign key is applied in Payment table			1			1			1			1
✓ Car Foreign key is applied in ParkingRecord table			1			1			1			1
✓ ParkingSlot Foreign key is applied in ParkingRecord table			1			1			1			1
Indicator 2.4: React basics are correctly applied based on application requirements												
✓ Function is declared			2			2			2			2
✓ Return method is included			1			1			1			1

Note: Write this candidate information



Candidate's Name	(1)			(2)			(3)			(4)		
		
		
		
Index number												
Starting Time												
Ending Time												
✓ Function is exported			1			1			1			1
✓ Component is mounted to the DOM			2			2			2			2
✓ JSX is used inside return method			1			1			1			1
✓ JSX is well structured			1			1			1			1
✓ Login form is created			1			1			1			1
✓ Car form is created			1			1			1			1
✓ ParkingSlot form is created			1			1			1			1
✓ ParkingRecord form is created			1			1			1			1
✓ Payment form is created			1			1			1			1
Indicator 2.5: UI navigation is correctly applied based on application requirements												
✓ Routes are configured			2			2			2			2
✓ Routing is performed			1			1			1			1
✓ Links are created			2			2			2			2
✓ Navigation between components is enabled			2			2			2			2
Indicator 2.6: Tailwind utility classes are correctly Applied based on UI Design												
✓ Tailwind CSS is installed			1			1			1			1
✓ Tailwind CSS is configured			1			1			1			1
✓ Tailwind is imported in the project			1			1			1			1
✓ Class elements are used in JSX			2			2			2			2

Note: Write this candidate information




Candidate's Name	(1)			(2)			(3)			(4)		
		
		
		
Index number												
Starting Time												
Ending Time												
✓ Flexbox is applied			1			1			1			1
Indicator 2.7: Responsive design principles are properly applied according to the required design												
✓ Flexible grid layouts are used			1			1			1			1
✓ Media breakpoints are removed			1			1			1			1
✓ Readability is Considered			1			1			1			1
✓ Interactive elements are applied			1			1			1			1
Indicator 2.8: Tailwind styles for a unique look are accurately customized based on tailwind configuration												
✓ Attractive colors are used			1			1			1			1
✓ Text formatting is performed			1			1			1			1
✓ UI components are attractive			1			1			1			1
✓ Hover state is used			1			1			1			1
✓ Focus state is used			1			1			1			1
Indicator 2.9: Server and database connections are properly established according to development environment												
✓ Server JS file is created			2			2			2			2
✓ Express.js package is imported			1			1			1			1
✓ Cors package is imported			1			1			1			1
✓ Port number is identified			1			1			1			1
✓ Express object is created			1			1			1			1

Note: Write this candidate information



Candidate's Name	(1)			(2)			(3)			(4)		
		
		
		
Index number												
Starting Time												
Ending Time												
✓ Listen method is applied			1			1			1			1
✓ MySQL/ MongoDB package is used			2			2			2			2
✓ Create connection method is used			1			1			1			1
✓ Host parameter is passed			1			1			1			1
✓ Username parameter is passed			1			1			1			1
✓ Password parameter is passed			1			1			1			1
✓ Database parameter is passed			1			1			1			1
✓ Messages are displayed			1			1			1			1
Indicator 2.10: RESTFUL APIs are effectively implemented based on backend functionalities												
✓ POST endpoint is created			2			2			2			2
✓ GET endpoint is created			2			2			2			2
✓ PUT endpoint is created			2			2			2			2
✓ DELETE endpoint is created			2			2			2			2
✓ Insert statement is used			2			2			2			2
✓ Select statement is used			2			2			2			2
✓ Update statement is used			2			2			2			2
✓ Delete statement is used			2			2			2			2
Indicator 2.11: API integration is properly implemented based on user requirements												
✓ API requests folder is created			2			2			2			2

Note: Write this candidate information




Candidate's Name	(1)	(2)	(3)	(4)
Index number				
Starting Time				
Ending Time				
✓ User API JS file is created		1	1	1
✓ Car API file is created		1	1	1
✓ ParkingSlot API file is created		1	1	1
✓ Payment API file is created		1	1	1
✓ ParkingRecord API file is created		1	1	1
✓ Axios is imported		1	1	1
✓ API client is set		1	1	1
✓ POST http method is used		2	2	2
✓ Base URL parameter is passed		2	2	2
Indicator 2.12: User Authentication, Authorization and Accountability (AAA) are carefully applied based on NPM Universal Access Control (UAC)				
✓ Form data is validated		2	2	2
✓ Password input is encrypted		1	1	1
✓ Duplicate data entry is eliminated		2	2	2
✓ Password recovery is implemented		2	2	2
Sub-total/111/111/111/111
Percentage (50)
3. Product presentation/Exhibition and Quality assessment (30%)	Indicator 3.1: Project is effectively presented			
✓ The product name is mentioned		1	1	1
✓ The key steps of the process are stated		3	3	3

Note: Write this candidate information



Candidate's Name	(1)			(2)			(3)			(4)		
		
		
		
Index number												
Starting Time												
Ending Time												
✓ The use/function/application/importance of the product is stated			2			2			2			2
✓ Challenges/difficulties met are stated			1			1			1			1
✓ Ways to overcome the challenges are stated			1			1			1			1
✓ Candidate voice is audible			1			1			1			1
✓ Body language is used			1			1			1			1
Indicator 3.2: Usability is correctly tested according to expected results												
✓ Input validation prevents invalid data			2			2			2			2
✓ Form submission saves data			2			2			2			2
✓ Data retrieval displays records			2			2			2			2
✓ Update functionality works			2			2			2			2
✓ Delete functionality works			2			2			2			2
Indicator 3.3: UI navigation is correctly applied based on application requirements												
✓ Navigation menus are interactive			1			1			1			1
✓ Buttons are interactive			1			1			1			1
✓ Fonts style is consistent			1			1			1			1
✓ Intuitive colors are used			1			1			1			1
✓ UI colors are consistent			1			1			1			1
✓ Bill is generated to the driver			3			3			3			3
✓ Generated bill is relevant			3			3			3			3

Note: Write this candidate information 	Candidate's Name		(1)		(2)		(3)		(4)		
				
				
				
	Index number										
	Starting Time										
	Ending Time										
	✓ Daily based parking sales report is generated				3			3			3
✓ Generated daily based parking sales report is relevant				3			3			3	
Sub-total	/37	/37	/37	/37			
Percentage (30)				
4. Closing activities (5%)	Indicator 4.1: Project configurations and its dependencies are properly removed										
	✓ Project folder is removed				1			1			1
	✓ Database is deleted				1			1			1
	✓ Global dependencies are removed				1			1			1
	✓ Text editor is not uninstalled				1			1			1
	✓ XAMPP is not uninstalled				1			1			1
	Indicator 4.2: Workplace laws and Time use are properly respected										
	✓ Paper is free of excessive erasures				1			1			1
	✓ Waste materials are disposed off				1			1			1
	✓ Unnecessary creases or tears are avoided on papers used.				1			1			1
	Indicator 4.3: Time is properly managed										
	✓ Time is respected				2			2			2
	Sub-total	/10	/10	/10	/10		
	Percentage (5%)			

3. COMPUTATION OF SCORES

FORMULA:

1. In the assessment forms, assessment criteria have specific indicators with elements to be checked. They are marked either Yes or No with Yes equivalent to Max score and No equivalent to 0 score.
2. **Points (P)** result in each **criterion (C)** is equal to the **sub-total (St)** of that criterion multiply the **weight of the criterion (Wc)** divided by **total max score (Ms)** of that criterion. The computation of points is up to two (2) decimal places.

$$P = \frac{St * Wc}{Ms}$$

Eg: St: 30: Ms: 35, Wc: 20 [Preliminary Activities Performance (20%)]

$$P = \frac{30}{35} * 20 = 17.14$$

St: 58: Ms: 72, Wc: 40 [Process and fulfillment of the task (40%)]

$$P = \frac{58}{72} * 40 = 32.22$$

etc..... for each criterion

3. The initial result of one assessor is equal to the sum of all criteria results.

Result = Points from PAP + Points from PFT + Points from PPQ + Points from CA

Where: **PAP:** Preliminary Activities Performance, **PFT:** Process and Fulfillment of the Task, **PPQ:** Product Presentation/Exhibition and Quality assessment, **CA:** Closing Activities

4. The final result (**FR**) of each candidate is the average of the results from the three assessors of the panel. FR is up to two (2) decimal places.

$$FR = (Ass1Res + Ass2Res + Ass3Res) / 3$$

For more details, find the appendix in rules and regulations

Score computation table

S/N	Criteria	Candidate's Name (1)	Candidate's Name (2)	Candidate's Name (3)	Candidate's Name (4)
		Points (P)	Points (P)	Points (P)	Points (P)
1	Preliminary Activities Performance: PAP (15%)				
2	Process and fulfillment of the task: PFT (50%)				
3	Product Presentation/ Exhibition and Quality assessment: PPQ (30%)				
4	Closing activities: CA (5%)				
Result/ 100					

Assessor:

Name	Institution	Qualification and Level	Telephone number	Signature

Note: All pages should have initial signature of the assessor

Done on, 2025 at (hour): Examinations center: