

Module Details		Learner's Details	
SECTOR:	ICT	Regno:	
SUB-SECTOR:	Information Technology (IT)	Class:	Level 6 IT (Year 1)
CERTIFICATE:	TVET Certificate VI in Information Technology	Trainer's Details	
Module (Code & Name):	ITLDF601 – Frontend development	Names:	SAFARI Cyprien
Competence:	Develop Frontend	Duration:	240min
Training Centre:	RP Karongi College	Date:	April 03, 2025
Type of evidence:	Performance evidence	Signature:	

PROJECT BASED PRACTICAL ASSESSMENT /100 marks

Objectives

This practical assessment assesses students' ability to develop a frontend for an **Online Ticket Reservation System** using React.js. The system should allow users to book, manage, and view ticket reservations efficiently.

Scenario

A company wants to develop an **Online Ticket Reservation System** where users can book tickets for various events, transportation, or shows. The system should allow users to search for available tickets, book tickets, and manage reservations.

General Requirements & Guidelines

- Students must work individually on the project.
- The project should be built using **React.js** with a proper component-based architecture.
- **React Router** must be implemented for navigation between pages.
- **Material UI** must be used for styling, including icons, tables, data tables (for automatic sorting, filtering, and pagination), and form components.
- **Recharts** should be used to visualize ticket booking statistics.
- The UI should include both **Dark Mode and Light Mode**, with a toggle button in the header using a half-moon icon.
- Additionally, **Dark and Light buttons** should be placed in the aside section for mode selection.
- Data tables should include **View and Delete** action buttons for managing reservations.
- Forms with image upload must display a preview of uploaded images (if applicable).
- Each section must include an **"Add New"** button for adding new ticket reservations.
- **Form validation must be implemented** to ensure data integrity and prevent empty or incorrect submissions.

- **Validation requirements:**

- Fields should not be left empty.
- Email fields must have a valid email format.
- Password fields must include a minimum of 8 characters, at least one uppercase letter, one number, and one special character.
- Image uploads should accept only specific file types (JPEG, PNG, etc.) and limit file size.

Technical Requirements

1. **Routing:** React Router must be implemented with proper navigation between pages (e.g., Home, Booking, Search, Ticket Details).
2. **Material UI:** Icons, tables, forms, and layout components should be used for consistency.
3. **Data Tables:** Sorting, filtering, and pagination should be handled automatically using Material UI for viewing reservations.
4. **Recharts:** Display analytics on ticket booking trends.
5. **Dark/Light Mode:**
 - Toggle button (half-moon icon) in the header.
 - Separate Dark and Light buttons in the aside section.
6. **Table Actions:**
 - View button to open a modal or separate page displaying detailed ticket information.
 - Delete button to remove a reservation with a confirmation prompt.
7. **Ticket Booking Features:**
 - Users can search for available tickets based on criteria.
 - Users can book tickets and view booking confirmations.
 - Users can browse and filter ticket reservations.
 - Ticket booking analytics displayed using Recharts.
8. **Admin Panel:**
 - Admins should be able to manage ticket reservations.
 - View analytics using Recharts.
9. **Registration Forms:**
 - Fields for user registration and ticket details.
 - Image upload with preview before submission (if applicable).
 - Add New button to add new ticket reservations.
 - **Form validation to prevent empty or incorrect data submission.**

Assessment Checklist

Category	Criteria	Marks	Yes	No
Project Structure & UI	Proper folder structure and component-based architecture	10		
	Responsive design implemented using Material UI	5		
	Dark and Light modes correctly toggle and persist	5		
Routing & Navigation	Navigation bar correctly links to all required pages	5		
	React Router properly handles routing between pages	5		
	Dynamic route handling for viewing ticket details	5		
Data Management & Tables	Data tables support sorting, filtering, and pagination	10		
	View button displays detailed ticket information	5		
	Delete button prompts confirmation before removal	5		
Ticket Booking Features	Users can search for available tickets	5		
	Users can book tickets and view booking confirmations	5		
	Users can browse and filter reservations	5		
	Ticket booking analytics displayed using Recharts	5		
Forms & Image Upload	Registration form includes all required fields	5		
	Image upload feature shows a preview before submission	5		
	"Add New" button successfully adds new ticket records	5		
	Form validation implemented to prevent incorrect input	5		
	Email validation ensures proper format	5		
	Password validation follows security criteria	5		
Code Quality & Submission	Well-structured and readable code	5		
	Proper use of React.js best practices	5		
Total Marks	100			