Assignment: Analysis of Functional and Non-Functional Structures

Project: Edura - Online Education Platform

Student Name: [Your Name]

Date: October 26, 2023

## 1. Introduction

This document provides a technical breakdown of the "Edura" website project. It identifies and categorizes the code components into Functional Structures (logic and interactivity) and Non-Functional Structures (static layout and visual design). The analysis is based on the source code provided in script.js, index.html, and style.css.

## 2. Functional Structures (Logic & Interactivity)

Definition: Functional structures are the active components of the website. They "do" work by handling user input, processing data, manipulating the DOM, or communicating with external APIs. In this project, these are primarily located in the JavaScript file.

Key Examples Found:

• AI Syllabus Generator (generateSyllabus)

Description: This function captures user input from the "topic" field, sends a request to the Gemini API, and dynamically renders a 4-week study plan into the HTML.

Type: API Integration / Dynamic Content Generation.

• Form Validation (handleFormSubmit)

Description: A generic handler for Login, Signup, and Contact forms. It intercepts the submit event, checks if required fields are filled, validates that passwords match (for signup), and simulates a loading state.

Type: Input Validation / Event Handling.

• Dashboard Visualization (renderDashboardCharts)

Description: Initializes the Chart.js library to render the "Learning Performance" (line chart) and "Course Category Distribution" (doughnut chart) based on hardcoded data arrays.

Type: Data Visualization.

• UI Utilities

toggleErrorPage(show): Controls the visibility of the full-screen error overlay.

togglePassword(inputId, icon): Switches the input type between 'password' and 'text' to reveal or hide characters.

3. Non-Functional Structures (Static Layout & Design)

Definition: Non-functional structures define the look, feel, and organization of the website. They do not execute logic but provide the necessary framework and visual styling. These are found in the HTML structure and CSS stylesheets.

Key Examples Found:

• Structural Framework (HTML)

Navigation Bar (): The fixed container at the top of the page holding the logo, menu links, and login button.

Hero Section: Contains the main headline and visual elements.

Footer: The static block at the bottom of the page containing site links and social icons.

• Layout Systems (CSS)

Bento Grid (.bento-grid): A CSS Grid layout for features.

Course Grid (.courses-grid): A responsive grid for course cards.

• Visual Styling (CSS)

CSS Variables (:root): Definitions for the site’s color palette.

Animations: float and reveal animations used for visual storytelling.

4. Summary Table

Category | Component Name | Description | File Source

Functional | generateSyllabus() | Fetches AI study plans from Gemini API | script.js

Functional | handleFormSubmit | Validates forms | script.js

Functional | renderDashboardCharts | Renders analytics using Chart.js | script.js

Functional | togglePassword | Toggles password visibility | script.js

Non-Functional | | Static navigation container | index.html

Non-Functional | .bento-grid | CSS grid for feature layout | style.css

Non-Functional | .hero | Main landing structure | index.html

Non-Functional | :root Variables | Defines global theme colors | style.css