

# CAPSTONE PROJECT

## DIGITAL PROBLEM SOLVER LAB

### Objective

Build a digital problem-solving system that collects data, analyses it accurately, applies logical rules through coding, and presents clear solutions for maths and science scenarios.

### Session 1: Structured Data Collection and Comparison Using Excel

#### Activities

##### 1. Designing a Controlled Data Sheet

###### ◊ Task

- ✦ Students will create an Excel worksheet to record values such as speed readings, temperature values, electricity units, or angle measurements using Form in Excel.

###### ◊ Skills Practised

- ✦ Using Excel Forms for data entry
- ✦ Applying data validation to restrict incorrect values
- ✦ Organising data using customised worksheet tabs

##### 2. Comparing and Summarising Data

###### ◊ Task

- ✦ Students will compare data sets using filters, conditional formatting, subtotal command, and pivot tables to identify maximum, minimum, and grouped results.

###### ◊ Skills Practised

- ✦ Filtering data for comparison
- ✦ Highlighting values using conditional formatting
- ✦ Summarising results using subtotals and pivot tables

#### Outcome for Session 1

- ◊ A validated and well-organised Excel workbook.
- ◊ Clear summaries showing comparisons and totals.



## Session 2: Planning Logic Using Flowcharts and Control Structures

### Activities

#### 1. Creating Flowcharts for Decision-Based Problems

##### ◊ Task

- ✦ Students will draw flowcharts for problems such as:
  - Odd or even number checking
  - Greatest number comparison
  - Identifying angle types
  - Electricity bill calculation

##### ◊ Skills Practised

- ✦ Using correct flowchart symbols
- ✦ Following flowchart structure and principles
- ✦ Understanding the advantages and limitations of flowcharts

#### 2. Translating Flowcharts into MakeCode Arcade Logic

##### ◊ Task

- ✦ Students will convert flowcharts into block-based logic using control structures, loops, jump statements, and functions.

##### ◊ Skills Practised

- ✦ Applying loops and conditional logic
- ✦ Using functions to organise logic
- ✦ Controlling program flow using jump statements

### Outcome for Session 2

- ◊ Well-designed flowcharts to represent the step-by-step logic of the program.
- ◊ Functional logic created using MakeCode Arcade to control game behaviour and actions.

## Session 3: Applying Coding Logic to App Planning

### Activities

#### 1. Solving Maths and Science Problems Using Coding Logic

##### ◊ Task

- ✦ Students will apply logical operators, relational operators, and if-else conditions to solve problems such as:
  - Speed and average speed



- Room temperature reading
- pH value classification
- Force calculation

#### ◊ Skills Practised

- ✦ Using conditional statements
- ✦ Applying mathematical operators logically
- ✦ Making decisions based on computed results

## 2. Designing a Simple Problem-Solving App Concept

#### ◊ Task

- ✦ Students will design an app idea that helps users calculate or classify values (maths or science based). They will define the app type, category, platform, and basic working.

#### ◊ Skills Practised

- ✦ Understanding what an app is
- ✦ Differentiating Android and iOS
- ✦ Identifying app categories and purpose

### Outcome for Session 3

- ◊ Correctly solved maths and science problems using coding logic.
- ◊ A clearly defined utility app concept.

### Final Deliverables

Tick (✓) the box if submitted:

- ◊ Excel workbook with validated data, filters, subtotals, and pivot tables.
- ◊ Flowcharts for logical, mathematical, and scientific problems.
- ◊ MakeCode Arcade programs based on control structures.
- ◊ A documented app concept for problem solving.



This project demonstrates how structured data handling, logical planning, coding techniques, and app concepts work together to solve real-world problems accurately and efficiently.

