SYLLABUS OF

Bachelor of Computer Application

PROGRAM

BCA Fifth Semester

Department of Computer Science & IT

Mangaldai College (Autonomous)

Semester-III			
Type	**Course Code**	**Course**	**T+P (Credit)**
Core	*BCA-DSC-5014*	**Software Engineering**	**4**
Core	*BCA-DSC-5024*	**Java Programming**	**3+1=4**
Core	*BCA-DSC-5034*	**Computer Networks**	**3+1=4**
Core	*BCA-DSC-5044*	**Information Security and Cyber Laws**	**4**
	BCA-DSC-5054	**UI/UX Design**	**3+1=4**
Internship	*BCA-DSC-5064*	**Internship**	**4**
			20

Course Information

Course Title	:	Software Engineering
Course Code	:	BCA-DSC-5014
Programme	:	Bachelor of Computer Applications (BCA)
Semester / Year	:	BCA 5th Semester
Credits	:	4
Contact Hours	:	Theory: 60 hrs. (60 classes)
Marks	:	**Theory: 100** Internal: 40 ESE : 60
Course Objectives	:	✓ To introduce fundamental concepts of Software Engineering and distinguish it from computer science and system engineering. ✓ To familiarize students with SDLC models, including traditional and agile approaches, and their applicability. ✓ To develop skills in requirement analysis, feasibility study, and preparation of SRS documents. ✓ To impart knowledge of software project management, including planning, estimation, scheduling, and documentation. ✓ To provide exposure to software design, coding standards, testing, reliability, and maintenance practices.
Course Outcomes (COs)	:	**CO1** Explain core Software Engineering concepts and evaluate characteristics of good software products, differentiating between programs and software systems. Analyze, compare, and select appropriate SDLC models, including agile and traditional approaches, for different types of software projects. **CO2** Perform requirement analysis, conduct feasibility studies, and prepare effective SRS documents using structured techniques such as decision trees, decision tables, and DFDs. **CO3** Apply software project management techniques to estimate size, cost, and effort; develop project schedules; and organize project teams using suitable management models. Design, code, test, and maintain software systems by applying design principles (cohesion and coupling), UML modeling, testing strategies, reliability metrics, and maintenance methodologies
Recommended Books & References	:	1. **Rajib Mall: Fundamentals of Software Engineering; PHI Learning Pvt. Ltd.** 2. **Roger S. Pressman: Software Engineering: A practitioner's Approach; McGraw Hill.**

Course Content / Units

A. Theory		Class	Marks
Unit 1:	**Introduction** Definition of Software Engineering, differentiation between Computer Science, Software Engineering and System Engineering, Program V/s software product, Exploratory style and modern style of software development, need of software engineering, characteristics of good software product	**7**	**7**
Unit 2:	**Software Development Life Cycle models** Definition of software development Life cycle (SDLC) models, Various life cycle modes: Classical Waterfall model, Iterative Waterfall model, Prototyping model, Evolutionary (Incremental) model, Spiral model, Agile Model, Agile V/s traditional SDLC Models, SCRUM model, Advantages and disadvantages of each of these SDLC models.	**15**	**15**
Unit 3:	**Requirement Analysis and Specification** What is Requirement Analysis and Gathering, Concept and Importance of Feasibility Study in Software design, Types of Feasibility: Technical, Economical and Operational feasibility, Software Requirement Specification (SRS) document, Components of an SRS (Software Requirement Specification): Functional and Non-Functional Component, Properties of a good SRS, Different users of SRS, Techniques to represent Complex Logic in SRS: Decision Tree and Decision Table.	**10**	**10**
Unit 4:	**Software Project Management** Basic idea of Software Project Management, Job Responsibilities of a Software Project Manager, Need of SPMP (Software Project Management Plan) document, Contents of SPMP, Need of Software documentation, Internal and External documentation, Software size estimation using Lines of Code (LOC), Merits and Demerits of LOC metric, Function Point Metric, 3D Function Point metrics, Project Estimation Techniques: Empirical estimation and Heuristics estimation techniques. Empirical estimation techniques: Delphi Cost Estimation and Delphi Cost Estimation. Heuristic Estimation Techniques: Basic COCOMO model and Intermediate COCOMO model. Project Scheduling: Work break down structure, Activity Networks and Critical Path Method.	**10**	**10**
Unit 5:	**Software Design principles and Methodology** Top down and bottom up approach, External Design, Architectural Design and Detailed design, Concept of Cohesion in software design, Classification of Cohesions, Basic concept of Coupling, Classification of Couplings, Introduction to software Analysis and Software Design (SA/SD), Introduction to Data Flow Diagram, Symbols used in DFD, Context Diagram in DFD, Advantages and Disadvantages of DFDs., Balanced DFD, Structured Design: Transaction Analysis and Transform Analysis. Need of Object Oriented Design and Analysis, UML (Unified Modeling Language), different views of UML, Various UML Diagrams: Use Case diagram, Class Diagram, Object Diagram, Sequence Diagram and Collaboration diagram	**10**	**10**

Unit 6:	**Coding and Testing** Goals of coding, Code Review techniques: Code Walkthrough, Code Inspection, Definition of Test cases, test suits, negative testing and positive testing. Different levels of software testing: *unit testing*, *Integration Testing*, *System Testing* and *acceptance testing*. Differentiation between Verification and Validation, Black box testing approaches: *Equivalent Class Partitioning* and *Boundary Value Analysis*, White Box testing approaches: *Statement Coverage*, *Branch Coverage, Condition* Coverage and Path *Coverage. Approach*, MaCabe's Cyclomatic Complexity, Basic idea of various system testing approaches: *Smoke testing, Stress testing, Volume testing* and *Compatibility testing.*	**5**	**5**
Unit 7:	**Software Reliability and Maintenance** What is reliability? Reliability metrics of Software Products: ROCOF, MTTF, MTTR, MTBF, POFOD and availability. ISO 9000 Certification, need of ISO Certification, How to get ISO 9000 certification, Definition of Software Maintenance, Types of Software maintenance: *Corrective, Adaptive* and *Perfective* maintenance, Estimation of Software Maintenance Cost.	**3**	**3**

Course Information

Course Title	:	Java Programming
Course Code	:	BCA-DSC-5024
Programme	:	Bachelor of Computer Applications (BCA)
Semester / Year	:	BCA 5th Semester
Credits	:	3+1=4
Contact Hours	:	Theory: 45 hrs. (15 classes)
Marks	:	**Theory: 100** Internal: 40 ESE : 60
Course Objectives	:	✓ Understand the fundamentals, features, and architecture of Java programming. ✓ Apply data types, operators, arrays, and control statements to develop Java programs. ✓ Implement object-oriented concepts such as classes, objects, inheritance, and polymorphism. ✓ Develop programs using strings, packages, exception handling, and file I/O. ✓ Design GUI-based applications using Swing and perform database connectivity using JDBC and MySQL.
Course Outcomes (COs)	:	**CO1** Explain the fundamentals, features, and architecture of Java programming. **CO2** Develop programs using data types, operators, arrays, and control statements. **CO3** Apply object-oriented programming concepts to develop Java applications. **CO4** Implement string handling, packages, exception handling, and file operations. **CO5** Develop GUI-based applications using Swing and database applications using JDBC and MySQL.
Recommended Books & References	:	**a) Java: The Complete Reference, Herbert Schildt, McGrawHill** **b) Java How to Program, Paul Deitel, Harvey Deitel, Pearson**

Course Content / Units

B. Theory		Class	Marks
Unit 1:	**Introduction** High level language, compiled and interpreted languages, history of java programming language, compilation of java code, bytecode, java interpreter, javac and java command, path environmental variable, Java IDE, features of java programming language: simple, object oriented, robust, architecture neutral and interpreted	**3**	**3**
Unit 2:	**Data types, operators and control statements** Java as strongly typed language, primitive data types, integer data types: byte, short, int and long, floating point data types: float and double, character data type, boolean data type, literals: integer literals, floating-point literals, boolean literals, character literals and string literals, declaring a variable, dynamic Initialization, the scope and lifetime of variables, type-casting in java, one dimensional array, multi-dimensional array, arithmetic operators: the basic arithmetic operators, the modulus operator, arithmetic compound assignment operators, increment operator and decrement operator, bitwise operators, relational operators, short circuit logical operator, the assignment operator, branching statements: if-else and switch-case statements, looping statements: while, do-while, for and for-each statements, jump statements: break and continue	**12**	**12**
Unit 3:	**Object oriented features of java** Defining a class, member variable and member methods, access specifiers: default, private and public, declaring objects, assigning object reference variables, constructors, parameterized constructors, the this keyword, garbage collection, the finalize( ) method, overloading methods, overloading constructor, static keyword, final keyword, command line arguments in java, inheritance, super class and sub class, protected access specifier, super keyword, constructor call in multilevel inheritance, method overriding, dynamic method dispatch, abstract class, interfaces, type wrappers	**10**	**10**
Unit 4:	**String handling and packages** String class, String constructors, String length, special string operations: string literals, string concatenation, string concatenation with other data types, string conversion and toString( ), character extraction: charAt( ), getChars( ), string Comparison: equals( ) and equalsIgnoreCase(), regionMatches( ), startsWith( ) and endsWith( ), equals( ) Versus ==, compareTo( ), searching strings, data conversion using valueOf( ), StringBuffer, StringBuffer constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ), package, defining a package, CLASSPATH, importing packages	**5**	**5**
Unit 5:	**Exception handling and I/O** Exception-handling, exception types, uncaught exceptions, try and catch block, multiple catch blocks, nested try statements, throw, throws, finally, java's built-in exceptions, creating own exception classes, java I/O classes, reading console input, writing console output, reading and writing files	**5**	**5**

Unit 6:	**Swing package and database connectivity** Swing package, simple GUI-Based Input/Output with JoptionPane, Jframe, JLabel, JTextField, Jbutton, handling event in a Jframe object, layout managers: BorderLayout, FlowLayout, GridLayout, CardLayout, GridBagLayout, JtoggleButton, JCheckBox, JRadioButton, Jlist, JcomboBox, JDBC, JDBC driver, connectivity steps, connectivity with MySQL, DriverManager class, Connection class, Statement class, ResultSet class, PreparedStatement class	**10**	**10**

B. Practical / Lab	**Class**	**Marks**
List of Practical's 1. Java programs to demonstrate the use of data types and operators 2. Java input through Scanner class and JOptionPane class 3. Java programs to demonstrate the use of control statements. 4. Java programs to demonstrate the use of classes, objects, visibility modes, constructors and destructor. 5. Java programs to demonstrate the use of inheritance and polymorphism. 6. Java programs to demonstrate the use of polymorphism. 7. Java programs to handle strings,Java programs implementing exception handling. 8. Demonstrating the use and creation of packages in java. 9. Java program with JFrame, JTextfield and JButton with event handling 10. Using JLabel, JTextArea and JPasswordField in java with event handling 11. Working with layout managers in JFrame 12. Using JCheckBox, JRadioButton and JComboBox in a JFrame 13. Connecting JFrame components to a DBMS***NB****: Presentation and assignment copy submission at the end of the course completion*	**30**	**15**

Course Information

Course Title	:	Computer Networks
Course Code	:	BCA-DSC-5034
Programme	:	Bachelor of Computer Applications (BCA)
Semester / Year	:	BCA 5th Semester
Credits	:	3+1
Contact Hours	:	Theory: 45 hrs. (45 classes)
Marks	:	**Theory: 75** Internal: 30 ESE : 45 **Practical: 25** Internal: 10 ESE : 15
Course Objectives	:	✓ Understand the fundamentals, components, types, and architectures of computer networks. ✓ Explain physical layer concepts, transmission media, encoding, modulation, and switching techniques. ✓ Understand data link layer protocols, error detection, error correction, and flow control mechanisms. ✓ Analyze network layer and transport layer protocols, including IP, TCP, UDP, and routing algorithms. ✓ Understand application layer services such as DNS, WWW, URL, HTTP, and email architecture.
Course Outcomes (COs)	:	**CO1:** Explain the fundamentals, components, types, and architectures of computer networks. **CO2:** Describe transmission media, encoding, modulation, multiplexing, and switching techniques. **CO3:** Apply data link layer protocols for error detection, error correction, and flow control. **CO4:** Analyze network and transport layer protocols, IP addressing, routing, TCP, and UDP. **CO5:** Explain application layer protocols and services such as DNS, WWW, HTTP, URL, and email.
Recommended Books & References	:	**a) B. A. Forouzan: Data Communications and Networking, Fourth edition, THM, 2007.** **b) A. S. Tanenbaum: Computer Networks, Fourth edition, PHI , 2002.**

Course Content / Units

A. Theory		Class	Marks
Unit 1:	**Introduction to Computer Networks** Data communication system and its components, Definition of network, Types of network, Network topologies, Network protocol, Layered network architecture, Overview of OSI reference model, Overview of TCP/IP protocol suite.	**5**	**5**
Unit 2:	**Physical Layer Communication** Analog and digital signal, Definition of bandwidth, Maximum data rate of a channel, Line encoding schemes, Transmission modes, Modulation techniques, Multiplexing techniques- FDM and TDM, Transmission media-Guided and Unguided, Switching techniques- Circuit switching, Packet switching, Connectionless datagram switching, Connection-oriented virtual circuit switching.	**10**	**10**
Unit 3:	**Data Link Layer Functions and Protocol** Definition of Framing, Framing methods, Error detection techniques, Error correction techniques, Flow control mechanisms- Simplex protocol, Stop and Wait ARQ, Go-Back-N ARQ, Point to Point protocol.	**10**	**10**
Unit 4:	**Multiple Access Protocol and Networks** Basics of ALOHA protocols, Basics of CSMA/CD protocols, Ethernet LANS, Connecting LAN and back-bone networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways	**5**	**5**
Unit 5:	**Networks Layer Functions and Protocols** Connection oriented vs Connectionless services, Definition of Routing, Routing algorithms, IP protocol, IP addresses, ARP, RARP	**8**	**8**
Unit 6:	**Networks Layer Functions and Protocols** Transport services, TCP vs UDP protocol, TCP connection establishment- Three way handshakes, TCP connection release	**4**	**4**
Unit 7:	**Networks Layer Functions and Protocols** Overview of DNS, Overview of WWW, URL, Email architecture, HTTP protocol	**3**	**3**

B. Practical / Lab	Class	Marks
List of Practical's	**30**	**15**

1. Implement the data link layer farming methods such as Bit Stuffing.
2. Study of different types of Network cables.
3. Study of network IP.
4. Connect the computers in Local Area Network.
5. Study of basic network command and Network configuration commands.
6. Configure a Network topology using packet tracer software.
7. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
8. Simulate and implement Stop and Wait protocol for noisy channel.
9. Simulate and implement Go-Back-N sliding window protocol.
10. Simulate and implement Selective Repeat sliding window protocol.
11. Simulate and implement Dijkstra Algorithm for shortest path routing.
12. Simulate and implement Distance vector routing algorithm***NB***: *Presentation and assignment copy submission at the end of the course completion*

Course Information

Course Title	:	Information Security and Cyber Laws
Course Code	:	BCA-DSC-5044
Programme	:	Bachelor of Computer Applications (BCA)
Semester / Year	:	BCA 5th Semester
Credits	:	4
Contact Hours	:	Theory: 60 hrs. (60 classes)
Marks	:	**Theory: 100** Internal: 40 ESE : 60
Course Objectives	:	✓ To introduce fundamental concepts of information security, including security principles, cryptographic techniques, and security mechanisms used to protect information systems. ✓ To develop an understanding of program and operating system security issues, vulnerabilities, threats, and protection mechanisms. ✓ To provide knowledge of database security concerns, including access control, inference problems, integrity, confidentiality, and availability of data. ✓ To familiarize students with network security threats, cyber-attacks, security architectures, secure communication protocols, and defensive mechanisms. ✓ To create awareness of cyber laws, cyber crimes, and the legal framework governing information technology and cyber security, with emphasis on the Information Technology Act, 2000.
Course Outcomes (COs)	:	**CO1:** Explain core security concepts such as confidentiality, integrity, availability, cryptography, encryption algorithms, digital signatures, and certificates. **CO2:** Identify and analyze program-level and operating system-level security vulnerabilities, threats, and appropriate control mechanisms. **CO3:** Evaluate database security issues and apply techniques for access control, inference control, data encryption, and protection of sensitive information.Analyze various network-based cyber attacks and implement suitable security controls, protocols, and defensive strategies. **CO4:** Interpret cyber laws and legal provisions related to cyber crimes, penalties, and offences under the Information Technology Act, 2000 and its amendments.
Recommended Books & References	:	Merkow, M., & Breithaupt, J.(2005) **Information Security Principles and Practices. 5th edition. Prentice Hall**. **Cryptography And Network Security Principles And Practice, Fourth or Fifth Edition,** William Stallings, Pearson Edition. **Cyber Law & Cyber Crimes**, Advocat Prashant Mali; Snow White publications, Mumbai

Course Content / Units

A. Theory		Class	Marks
Unit 1:	**Introduction** Basic components of security (Confidentiality, Integrity and Availability), Attacks, Computer Crime, Security Services, Security Mechanism, Cyber Crimes, information Technology ACT, Cryptography, Substitution Cipher, Transposition Cipher, Block Cipher, Stream Cipher, Confusion, Diffusion, Symmetric Key, Asymmetric Key, Encryption, DES Algorithm, Hash Function, Digital Signature, Digital Certificate.	**10**	**10**
Unit 2:	**Program Security** Program Security, Program Errors, Buffer Oveflow, Incomplete mediation, Time-of-check to Time- of- use Errors, Malicious codes, Virus, Threats, Control against Programs, Program Security Issues. Protection in OS: Memory and Address protection, Access control, File protection, User Authentication	**15**	**15**
Unit 3:	**Database Security** Reliability, Integrity, Sensitive Data, Inference, Multilevel Security,Issues regarding the right to access information: Protecting Data, Multiple security level and categorization of data and users, Loss of integrity, Loss of availability, Loss of confidentiality, Access control, Inference control, flow control, data encryption	**10**	**10**
Unit 4:	**Security in Networks (Cyber Attack)** Threats in Networks, Security Controls- Architecture, Encryption, Content Integrity, Strong Authentication, Firewalls: Design and Types of Firewalls, Intrusion Detection System, Secure Email, Denial-of-service attacks, Man in the middle Attack, Phishing, Spoofing and Spam Attacks, Drive-by attack, SQL Injection, Birthday attack, Social Engineering attack, Password Attack. Cross site scripting Attack, Malware Attack, Administering Security, Security Planning, Risk Analysis, Organisational Security Policy, Web Servers and Browsers, HTTP, Cookies, Caching, Secure Socket Layer (SSL), Secure Electronic Transaction (SET), E-mail Risks, Spam, E-mail Protocols, Simple Mail Transfer Protocol (SMTP), Post office Protocol (POP), Internet Access Message protocol (ICMP), Secured Mail: Pretty Good Privacy (PGP), S/MIME (Secure/Multipurpose Internet Mail Extensions)	**12**	**12**
Unit 5:	**Cyber Laws** Cyber crime, Types of crimes, Information technology Act 2000: Salient Feature of IT Act 2000, various authorities under IT Act and their powers, Penalties & Offences, amendments, Sections under the Information Technology Act such as: ● [Section 43] Penalty and compensation for damage to computer etc. ● [Section 65] Penalty for temping with the computers source documents ● [Section 66] Punishment for hacking with computer system, data alteration etc	**13**	**13**

- [Section 66A] Punishment for sending offensive messages through any communication services
- [Section 66B] Receiving stolen computer's resources or communication devices dishonestly
- [Section 66C] Punishment for identity theft
- [Section 66D] Punishment for cheating by impersonation by using computer resource
- [Section 66E] Punishment for violation of privacy
- [Section 66F] Punishment for cyber terrorism
- [Section 67] Punishment for publishing or transmitting obscene material in electronic form
- [Section 67A] Punishment for publishing or transmitting of material containing sexually explicit act, etc. in electronic form
- [Section67B] Punishment for publishing or transmitting of material depicting children in sexually explicit act, etc. in electronic form
- [Section 72] Breach of confidentiality and privacy

Course Information

Course Title	:	UI/UX Design
Course Code	:	BCA-DSC-5054
Programme	:	Bachelor of Computer Applications (BCA)
Semester / Year	:	BCA 5th Semester
Credits	:	3+1
Contact Hours	:	Theory: 45 hrs. (45 classes) Practical 30 hrs.
Marks	:	**Theory: 75** Internal: 30 ESE : 45 **Practical: 25** Internal: 10 ESE : 15
Course Objectives	:	✓ To introduce students to the fundamental concepts of User Interface (UI) and User Experience (UX) design and their importance in modern web and mobile applications. ✓ To develop an understanding of user-centered design principles through user research, personas, user journeys, and problem definition. ✓ To equip learners with core UI design skills, including color theory, typography, layout design, and accessibility considerations. ✓ To provide hands-on experience in wireframing, designing, and prototyping using industry-standard tools such as Figma. ✓ To enable students to apply UI/UX concepts in real-world projects, conduct usability testing, and iteratively improve designs based on user feedback.
Course Outcomes (COs)	:	**CO1:** Clearly differentiate between UI and UX and explain the role of UI/UX design in enhancing user satisfaction and product usability. **CO2:** Conduct basic user research activities, create user personas, user stories, and user journey maps to inform design decisions.Design visually consistent and accessible user interfaces using appropriate color palettes, typography, layouts, and UI components. **CO3:** Create low-fidelity wireframes, high-fidelity screens, and interactive prototypes using Figma for both mobile and web application. **CO4:** Develop and present a complete UI/UX project, perform basic usability testing, analyze feedback, and refine designs accordingly.
Recommended Books & References	:	**Design Thinking for Beginners,** *R. Venkatesan, Notion Press* **User Experience Design: A Practical Guide.***Saurabh Tiwari,BPB Publications* **Figma for UI/UX Designers,** *Amit Gupta, BPB Publications*

Course Content / Units

A. Theory		Class	Marks
Unit 1:	**Introduction to UI & UX** Meaning of UI (User Interface) and UX (User Experience), Difference between UI and UX Importance of UI/UX in web and mobile applications, Role of a UI/UX designer, Examples of good and bad UI/UX designs, Overview of the design thinking process.	**10**	**10**
Unit 2:	**User Research & Wireframing (Hands-on)** Creating user personas (practical examples), Writing user problems and user stories, creating user journey maps, Paper wireframing and digital wireframing, Hands-on: Create low-fidelity, wireframes for a simple app, Tool use: Wireframing directly in Figma	**10**	**10**
Unit 3:	**UI Design Fundamentals** Choosing color palettes for real projects, Typography selection for readability, Designing buttons, forms, cards, and navigation, Layout design using grids, Hands-on: Design login, signup, and dashboard screens, Applying accessibility basics in design.	**10**	**10**
Unit 4:	**Designing & Prototyping in Figma** Creating frames for mobile and web, Using components and variants, Auto-layout for responsive design, Creating interactive prototypes, Linking screens with interactions, Hands-on: Build a clickable prototype of an app	**10**	**10**
Unit 5:	**Real-World Project & Usability Testing** Complete UI/UX project (individual or group) Project ideas: ● College website ● Wallet / expense tracker app ● Food delivery or learning app Conducting basic usability testing, Improving design based on feedback, Final project presentation and documentation	**5**	**5**

B. **Practical / Lab**	**Class**	**Marks**
List of Practical's	**30**	**15**

1. Introduction to Figma Workspace
    - Create a Figma account
    - Explore canvas, frames, layers, assets, and properties panel
2. UI Analysis of an Existing App
    - Select any popular website or mobile app
    - Identify good and bad UI/UX elements
    - Submit screenshots with short analysis
3. User Persona Creation
    - Create at least 2 user personas
    - Define user goals, problems, and behavior
4. User Journey Mapping
    - Create a user journey for a simple task (login, booking, payment)
    - Show steps, emotions, and pain points
5. Low-Fidelity Wireframing
    - Design paper or digital wireframes
    - Create low-fidelity wireframes in Figma
6. Design Color Palette & Typography
    - Create a color palette for a project
    - Select typography for headings and body text
7. Button & Form Design
    - Design buttons (primary, secondary, disabled)
    - Design a form with input fields and validation states
8. Login & Signup Screen Design
    - Design login screen
    - Design signup screen
    - Apply consistency and spacing rules
9. Dashboard UI Design
    - Design a dashboard layout
    - Use cards, charts placeholders, and navigation
10. Components & Auto Layout
    - Create reusable components
    - Apply auto layout for responsiveness Usability Testing
    - Conduct usability testing with at least 3 users
    - Note problems and suggestions
    - Improve UI based on feedback
11. Final Mini Project
    - Design a complete UI/UX project
    - Example: College website / Wallet app / Learning app
    - Create wireframes, UI screens, and prototype
    - Submit Figma link and brief project report

***NB**: Presentation and assignment copy submission at the end of the course completion*