



AVERIXIS SOLUTION

WEB DEVELOPMENT

MENTORSHIP PROGRAM WITH INTEGRATED LCNC

Certification Partner



OUR EXPERT MENTOR PANEL FROM



Starting Point For Your Career Path

Our Mission & Vision

We help undergrad and post grad students struggling to get industrial experience with our Industry Grade Mentorship programs which help them to become corporate-ready individuals and possess the skillset to take on any challenges without any self-doubt.



Mission

To transform the way people learn and develop their skills by providing a dynamic and immersive upskilling platform that delivers hands-on learning and practical industry experience, empowering learners to achieve their full potential and thrive in the rapidly changing world of work.



Vision

To be the leading provider of hands-on upskilling solutions that connect students with the best industry experts and provide them with real-world industry projects to prepare them for success in their chosen careers.

Why Averixis Adopted LCNC(Low Code No Code)

Freshers, college students and the people with no coding knowledge can now build apps, websites on their own with the help of LCNC. This feature helps you discover the uncovered areas and boost your confidence even if you don't have any coding knowledge.

Feed your creativity hunger and come up with a faster and the most effective project completion ways with

India's No. 1 LCNC integrated curriculum.

Why Startups are Betting Big on Low-Code/No-Code

BY: SAQIB JAN on february 2, 2024

It is exhaustive — from infrastructure to app delivery, from data to applications — to modernize your practices, processes and providers to ensure you have the underlying foundation to take advantage of whatever comes next.

Two or three years ago, apps created through low-code/no-code platforms were not usually as detailed under the surface as software developed from scratch, yet they sufficed for certain purposes. There was even a clear distinction between software developers and everyone else out of necessity because software development was incredibly difficult to master.

But now, as we head towards more advanced AI, the SaaS-based low-code/no-code (LCNC) platforms empower businesses to create software exponentially faster and cheaper than a code-based approach.

PUBLISHED IN



Building No- and Low-Code Tools into Your Workflow

BY: Nick Kolakowski on Jun 6, 2024

The idea of “citizen developers” with little coding experience using no- and low-code platforms to build apps isn’t a new concept; for many years, companies like Microsoft have released tools designed to empower pretty much anyone to produce mobile apps, games, and more.

While the idea of democratizing app-building is appealing to many, IT specialists and cybersecurity experts have long feared the not-so-controlled chaos that no- and low-code platforms could unleash within an organization with no guardrails in place.

The advent of generative AI may only heighten these fears, especially if employees rely on AI tools from outside their company’s sanctioned tech stack to build things (a trend cheekily known as ‘Bring Your Own Artificial Intelligence,’ or BYOAI).

But the fact is, no- and low-code tools will likely become more powerful in the years ahead, and

PUBLISHED IN



MONTH 01

WEEK 01

DAY

01

- ◆ Introduction to Web Development
- ◆ Overview of web development
- ◆ Key technologies: HTML, CSS, JavaScript
- ◆ Differences between front-end and back-end development

DAY

02

- ◆ Basic HTML Document Structure and Syntax
- ◆ HTML elements and attributes
- ◆ Structuring a basic HTML document
- ◆ Basic CSS Syntax
- ◆ CSS selectors, properties, and values
- ◆ Linking CSS to HTML

DAY 03

- ◆ Advanced HTML Tags and Attributes
- ◆ Forms, tables, lists
- ◆ HTML5 semantic elements (header, footer, section)
- ◆ Semantic HTML
- ◆ Importance of semantic HTML for SEO and accessibility

DAY 04

- ◆ CSS Layout Basics
- ◆ Box model (margin, border, padding, content)
- ◆ CSS positioning (static, relative, fixed, sticky)
- ◆ Responsive Design with Media Queries
- ◆ Media query syntax
- ◆ Creating responsive layouts

DAY 05

- ◆ Building a Simple Website
 - ◆ Project setup and folder structure
 - ◆ Writing HTML, CSS, and basic JavaScript
 - ◆ Testing and debugging
- **Live Project 1: Personal Portfolio Website**

WEEK 02

DAY
06

- Introduction to JavaScript ◆
- JavaScript basics: syntax, comments ◆
- Variables and data types (number, string, ◆
boolean, array, object)
- Operators (arithmetic, assignment, ◆
comparison, logical)

DAY
07

- Functions and Control Structures ◆
- Defining and invoking functions ◆
- Control structures: if-else, switch, loops ◆
(for, while, do-while)
- Debugging Tools and Techniques ◆
- Console and debugging tools in browsers ◆
- Common debugging practices ◆

DAY
08

- DOM Manipulation with JavaScript ◆
- Understanding the Document Object Model (DOM) ◆
- Selecting and manipulating DOM elements ◆
- Events and Event Listeners ◆
- Adding event listeners ◆
- Handling different types of events ◆
(click, input, submit)

DAY
09

- Creating Interactive Web Pages ◆
- Enhancing interactivity with JavaScript ◆
- Basic animations and effects ◆
- Introduction to jQuery ◆
- jQuery syntax and usage ◆
- Simplifying DOM manipulation with jQuery ◆

DAY
10

- Deploying a Website to a Web Server ◆
- Hosting options and domain registration ◆
- FTP and using platforms like GitHub Pages ◆
- Introduction to Version Control with Git and GitHub ◆
- Git basics: init, add, commit, push ◆
- Using GitHub for version control and collaboration ◆

Live Project 2: Interactive To-Do List Application •

WEEK 03

DAY

11

- ◆ Introduction to Backend Development
- ◆ Understanding server-side programming
- ◆ Overview of languages (PHP, Python, Ruby, Node.js)

DAY

12

- ◆ Basic Server Setup and Configuration
- ◆ Setting up a local development environment
- ◆ Introduction to web servers (Apache, Nginx)
- ◆ Handling HTTP Requests and Responses
- ◆ Understanding HTTP methods (GET, POST, PUT, DELETE)
- ◆ Basics of handling requests and responses

DAY 13

- ◆ Introduction to Databases
- ◆ Difference between relational and non-relational databases
- ◆ Overview of SQL and NoSQL databases

DAY 14

- ◆ SQL Basics
- ◆ Writing basic SQL queries (SELECT, INSERT, UPDATE, DELETE)
- ◆ Understanding joins and relationships between tables
- ◆ Setting Up a Database Server
- ◆ Installing and configuring a database server (e.g., MySQL, PostgreSQL)

DAY 15

- ◆ Connecting a Web Application to a Database
 - ◆ Database connections using server-side languages
 - ◆ Performing CRUD operations
-
- **Live Project 3: Basic Blog Platform**

WEEK 04

DAY
16

- Introduction to Object-Relational Mapping (ORM) Tools ◆
- Overview of ORM concepts ◆
- Examples of ORM tools (SQLAlchemy for Python, ActiveRecord for Ruby) ◆
- Introduction to Web Frameworks ◆
- Overview of popular frameworks (Flask, Django, Ruby on Rails, Express.js) ◆

DAY
17

- MVC Architecture ◆
- Understanding the Model-View-Controller pattern ◆
- How MVC is implemented in web frameworks ◆
- Setting Up a Web Application with a Framework ◆
- Creating a basic application using a framework (e.g., Flask) ◆

DAY
18

- Basic Routing and URL Handling ◆
- Defining routes and handling URL parameters ◆
- Creating dynamic routes ◆

DAY
19

- User Authentication and Authorization ◆
- Implementing user login and registration ◆
- Managing user sessions and roles ◆
- Building a Web Application with a Framework ◆
- Developing a complete feature with a framework ◆

DAY
20

- Handling State with Redux or Other State ◆
- Management Libraries
- Introduction to state management ◆
- Using Redux for state management ◆
in web applications
- Using Third-Party Libraries and Plugins ◆
- Integrating libraries and plugins to ◆
enhance functionality

Live Project 4: User Authentication System •

MONTH 02

WEEK 05

DAY

21

- ◆ Introduction to APIs and Web Services
- ◆ Understanding APIs and web services
- ◆ RESTful API design principles

DAY

22

- ◆ Building a Simple API with a Framework
- ◆ Creating endpoints and handling requests
- ◆ Testing and documenting APIs
- ◆ Consuming an API with JavaScript
- ◆ Making API calls with Fetch or Axios
- ◆ Handling API responses

DAY 23

- ◆ Introduction to Front-End Frameworks
- ◆ Overview of React, Angular, Vue.js
- ◆ Setting up a development environment

DAY 24

- ◆ Components and Props
- ◆ Understanding component-based architecture
- ◆ Passing data with props
- ◆ Routing and Navigation
- ◆ Implementing routing with front-end frameworks
- ◆ Managing navigation between components

DAY 25

- ◆ Building a Web Application with a Front-End Framework
- ◆ Developing a simple application using React, Angular, or Vue.js
- ◆ Project Application Session 1
- ◆ Applying learned concepts to a practical project
- **Live Project 5: RESTful API and Front-End Integration**

WEEK 06

DAY
26

- Handling State with Redux or Other State ◆
- Management Libraries
- Advanced state management techniques ◆
- Implementing Redux in a complex application ◆

DAY
27

- Introduction to Testing and Debugging ◆
- Web Applications
- Importance of testing in web development ◆
- Writing unit tests with Jest or other frameworks ◆

DAY
28

- Debugging Techniques and Tools ◆
- Effective debugging strategies ◆
- Using browser developer tools ◆

DAY
29

- Building a Responsive and Accessible Web Application ◆
- Best practices for responsive design ◆
- Accessibility guidelines and techniques ◆

DAY
30

- Responsive Design Techniques for Mobile Devices ◆
- Optimizing web applications for mobile ◆
- Using media queries and flexible layouts ◆

Live Project 6: Responsive and Accessible Web Application •

WEEK 07

DAY
31

- ◆ Introduction to Generative AI in Web Development
- ◆ Overview of AI and machine learning concepts
- ◆ Applications of AI in web development

DAY
32

- ◆ Using Generative AI for Content Creation
- ◆ Introduction to OpenAI and GPT models
- ◆ Creating AI-generated content for websites

DAY 33

- ◆ Prompt Engineering Basics
- ◆ Understanding prompt engineering
- ◆ Crafting effective prompts for AI models
- ◆ Integrating AI-Generated Content into Web Applications
- ◆ Using AI APIs to enhance web applications

DAY 34

- ◆ Introduction to No-Code Tools
- ◆ Overview of no-code platforms (e.g., Webflow, Bubble)
- ◆ Building a web application without coding



DAY 35

- ◆ Outcome-Driven Project with No-Code Tools
 - ◆ Developing a complete project using no-code tools
 - ◆ Showcasing the final project
-
- **Live Project 7: AI-Powered Blog Generator**

WEEK 08

DAY
36

- Advanced Prompt Engineering ◆
- Advanced techniques for prompt engineering ◆
- Customizing AI responses for specific use cases ◆
- AI-Driven User Experience Personalization ◆
- Using AI to create personalized user experiences ◆
- Implementing personalization in web applications ◆

DAY
37

- Using AI for Data Analysis and Visualization ◆
- Analyzing data with AI ◆
- Creating visualizations with AI tools ◆
- Integrating AI APIs into Web Applications ◆
- Using AI APIs to enhance functionality ◆
- Examples of popular AI APIs (e.g., Google Cloud AI, IBM Watson) ◆

DAY
38

- No-Code Tools for E-Commerce Applications ◆
- Building an e-commerce site with no-code platforms ◆
- Managing products, orders ◆

Our Alumni Work At



SUPPLEMENTARY PERKS



Resume Building Session



Our Courses Give You Hands On Experience With
Mock Interviews

Scroll Down For Contact Details



Dont Hesitate To Contact us!

Averixis Solution

STARTING POINT FOR YOUR CAREER PATH



www.averixis.com



+91-843111080

Follow us

