SOFTCRAYONS®

Softcrayons

PYTHON FULL STACK

Empowering minds shaping futures

www.softcrayons.com | +91 854 501 2345 | info@softcrayons.com

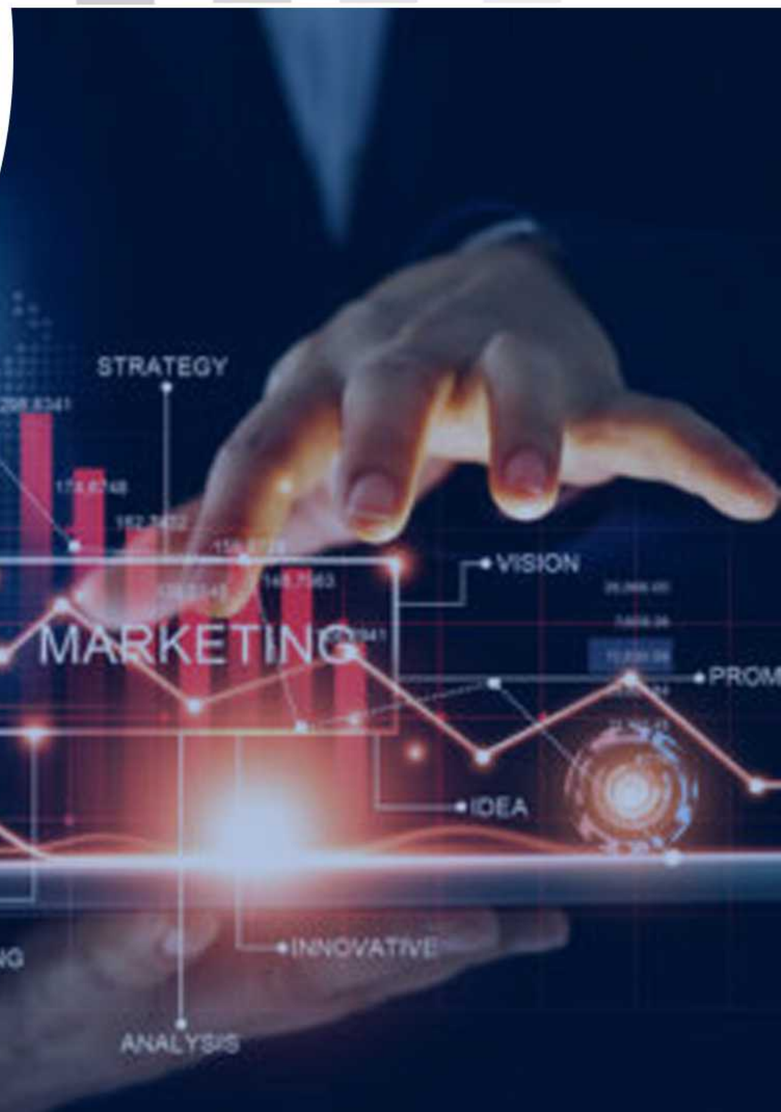
www.softcrayons.com

SOFTCRAYONS®

PROFESSIONAL CERTIFICATION IN

PYTHON FULL STACK ...

PROSPECTUS



SOFTCRAYONS®

Key Advantages of Choosing Softcrayons

FOR STUDENTS

Supplemental Learning Resources: Softcrayons offers offline and online courses, educational materials, and additional resources that can complement and enhance college students' learning.

Skill Development: Softcrayons offer courses and certifications focused on developing specific skills that are in high demand in the job market.

Career Exploration: Softcrayons offer a wide range of courses across various discplines, enabling college students to explore different fields and potential career paths.

Industry Relevance: Softcrayons frequently collaborates with industry professionals and experts to ensure that the knowledge and skills imparted are relevant and aligned with current industry practices and trends.

FOR FREELANCER & JOB SEEKER

Flexibility: Softcrayons offers online courses and programs that can be accessed from anywhere.

Skill Development: To acquire in-demand skills according to the latest industry trends and technologies to stay competitive in the job market.

SOFTCRAYONS®

Key Advantages of Choosing Softcrayons

Certifications: Softcrayons provides you with Professional Certifications and helps you with Resume Enhancement.

Career Support: Softcrayons also offers career counseling and job placement assistance, which can be invaluable for freelancers seeking new projects or job seekers looking for employment.

FOR ENTREPRENEURS AND BUSINESS OWNER

Upskilling and Reskilling: As the business landscape evolves rapidly, Softcrayons ensure that you stay up-to-date with the latest trends, technologies, and best practices.

Flexible Learning: Online courses offered by Softcrayons allow you to learn at your own pace, fitting your studies around your busy schedules.

Entrepreneurial Skills Development: Softcrayons offers courses and programs specifically designed to help entrepreneurs develop essential skills.

Cost-effective: Softcrayons provides more affordable learning options that help you invest in your professional development without straining your budget.

About The Program

This course is designed to provide a comprehensive understanding of Python programming and its application in full-stack web development. Students will learn Python fundamentals, backend development with frameworks like Django or Flask, front-end technologies like HTML, CSS, and JavaScript, and database management. The course will also cover deployment, testing, and other essential skills required for building complete web applications.

info@softcrayons.com

www.softcrayons.com

INDEX

1 HTML (HYPER TEXT MARKUP LANGUAGE):

- Introduction to HTML
- Basic HTML Structure
- Text Formatting
- Lists
- Links
- Images
- Tables
- Forms
- Multimedia
- Transitioning to HTML5
- Basic HTML5 Structure
- HTML5 Best Practices
- Integrating with CSS3 and JavaScript

2 CSS (CASCADING STYLE SHEETS):

- Introduction to CSS
- CSS Selectors
- CSS Box Model
- CSS Colors
- CSS Backgrounds
- CSS Text
- CSS Layout
- CSS Transitions
- CSS Transformations
- CSS Animations
- CSS Responsive Design
- CSS Frameworks, Preprocessors & Tools
- CSS Best Practices

INDEX

3 JAVASCRIPT

- Introduction to JavaScript
- JavaScript Basics
- DOM Manipulation
- JavaScript Objects
- JavaScript Data Structures
- Asynchronous JavaScript
- ES6 and Beyond
- Functions and Functional Programming
- DOM Events and Browser APIs
- JavaScript Tooling and Ecosystem
- Advanced JavaScript Concepts

4 BOOTSTRAP

- Introduction to Bootstrap
- Bootstrap Grid System
- Bootstrap Content
- Bootstrap Components
- Building Responsive Designs with Bootstrap
- Customizing Bootstrap with Sass

5 TAILWIND CSS

- Introduction to Tailwind CSS
- Tailwind CSS Fundamentals
- Layout and Flexbox
- Grid and Box Alignment
- Typography and Backgrounds
- Borders, Effects, and Transitions

INDEX

- Forms and Interactive Elements
- Customizing Tailwind CSS
- Integrating Tailwind CSS

6 REACT.JS

- Getting Started with React
- React Fundamentals
- React Components
- React State Management
- React Hooks
- React Router
- React Forms and Validation
- React and API Integration
- State Management with Redux
- Advanced React Concepts
- React Testing
- React Ecosystem and Tooling
- React for Mobile Development
- Real-World React Projects

ANY ONE DATABASE (MONGODB OR MYSQL)

7 MONGODB

- Introduction to NoSQL and MongoDB
- MongoDB Installation and Setup
- MongoDB Data Model
- CRUD Operations

INDEX

- Indexing and Performance
- Aggregation Framework
- Data Modeling and Schema Design
- Replication and Sharding
- MongoDB Integration
- Administration and Security
- Advanced Topics (Optional)

8 MYSQL

- Introduction to Databases and SQL
- MySQL Installation and Setup
- SQL Basics
- SQL Queries and Operators
- Database Design and Normalization
- Advanced SQL Topics
- MySQL Administration
- MySQL Programming
- Advanced MySQL Topics (Optional)

9 PYTHON LANGUAGE

- Introduction to Python
- Python Operators and Expressions
- Python Collections (Data Structures)
- Python Control Structures
- Python Functions
- Python Exception Handling
- Python Modules and Packages
- Python File Handling

INDEX

- Object-Oriented Programming in Python
- Python Functional Programming
- Python Testing and Debugging

10 DJANGO FRAMEWORK:

- Introduction to Django
- Django Basics
- Models and Databases
- Forms and User Input
- User Authentication
- Advanced Django Topics
- Third-Party Packages
- Testing and Debugging
- Django RESTful APIs
- Django and Front-end
- Other Important Topics

11 GIT AND GITHUB

- Introduction to Version Control
- Git Basics
- Branching and Merging
- Remote Repositories
- Collaborative Development

12 DEPLOYMENT AND DEVOPS

13 FINAL PROJECT

PYTHON FULL STACK

TRAINING CURRICULUM

1 HTML (Hyper Text Markup Language)

– Introduction to HTML

- **What is HTML?**
- **HTML Structure and Elements**
- **HTML Various Versions**

– Basic HTML Structure

- **HTML document structure**
- **HTML tags and elements**
- **HTML attributes**
- **HTML comments**

– Text Formatting

- **Headings (h1 - h6)**
- **Paragraphs (p)**
- **Line breaks (br)**
- **Horizontal rules (hr)**
- **Formatting text (b, i, u, mark, small, sub, sup etc)**

– Lists

- **Unordered lists (ul, li)**
- **Ordered lists (ol, li)**
- **Definition lists (dl, dt, dd)**

– Links

- **Anchor links (a)**
- **Linking to external pages**
- **Linking within the same page**
- **Email links**

– Images

- **Inserting images (img)**
- **Image attributes (src, alt, width, height)**

– Tables

- **Creating tables (table, tr, th, td)**
- **Table structure and layout**
- **Table attributes (border, cellpadding, cellspacing, etc.)**
- **Table headers and captions**

– Forms

- **Creating forms (form)**
- **Form elements (input, textarea, select, option, etc.)**
- **Form attributes (action, method, name, value, etc.)**
- **Form validation (client-side and server-side)**

– Multimedia

- **Video (video, source)**
- **Audio (audio, source)**
- **Embedding multimedia (object, embed)**
- **Media attributes (src, controls, width, height, etc.)**

– Transitioning to HTML5

- **Introduction to HTML5**
- **New elements and features in HTML5**
- **Differences between HTML4 and HTML5**

– Basic HTML5 Structure

- **HTML5 document structure**
- **HTML5 semantic elements (header, nav, section, article, aside, footer)**
- **HTML5 structural elements (div, span)**
- **HTML5 metadata elements (head, title, meta, link, style)**

– HTML5 Best Practices

- **HTML5 syntax and coding standards**

- **HTML5 validation and debugging**
- **Cross-browser compatibility**
- **Performance optimization**

– Integrating with CSS3 and JavaScript

- **Styling HTML5 elements with CSS3**
- **Manipulating HTML5 elements with JavaScript**

2 CSS (Cascading Style Sheets)

– Introduction to CSS

- **What is CSS?**
- **CSS syntax and structure**
- **Ways to include CSS (inline, internal, external)**

– CSS Selectors

- **Type selectors**
- **Class selectors**
- **ID selectors**
- **Universal selector**
- **Attribute selectors**
- **Combinators (descendant, child etc.)**
- **Pseudo-classes and pseudo-elements**

– CSS Box Model

- **Content, padding, border, and margin**
- **Box-sizing property**
- **Margin collapsing**

– CSS Colors

- **Pre-Defined Colors Name**
- **Hexadecimal color codes**
- **RGB and RGBA colors**
- **HSL and HSLA colors**
- **Opacity and alpha transparency**

– CSS Backgrounds

- **Background color**
- **Background image**
- **Background repeat, position, attachment, and size**
- **Multiple backgrounds & Gradients**

– CSS Text

- **Font families**
- **Font size and weight**
- **Text color and decoration**
- **Text alignment and indentation**
- **Line height and spacing**
- **Text transforms and shadows**

– CSS Layout

- **Display property (block, inline, inline-block)**
- **Positioning (static, relative, absolute, fixed, sticky)**
- **Floats and clearing**
- **Flexbox layout**
- **CSS Grid layout**

– CSS Transitions

- **Transition properties**
- **Transition timing functions**
- **Transition delays**

– CSS Transformations

- **2D transformations (translate, rotate, scale, skew)**
- **3D transformations**
- **Transform origin**

– CSS Animation

- **@keyframes rule**
- **Animation properties**
- **Animation timing functions**

– CSS Responsive Design

- **Media queries**
- **Responsive units (vw, vh, rem)**
- **Responsive images and media**
- **Responsive typography**

– CSS Frameworks, Preprocessors & Tools

- **Introduction to CSS frameworks (Bootstrap, Tailwind CSS etc.)**
- **Introduction to CSS preprocessors (Sass, Less)**
- **CSS Tools like: linters and formatters**

– CSS Best Practices

- **CSS syntax and formatting**
- **CSS selectors and specificity**
- **CSS organization and methodologies**
- **CSS optimization and performance**

3 JavaScript

– Introduction to JavaScript

- **Overview of JavaScript and its history**
- **Understanding the role of JavaScript in web development**
- **Embedding JavaScript in HTML and running it in a web browser**

– JavaScript Basics

- **Variables, data types, and type conversion**
- **Operators (arithmetic, assignment, comparison, logical)**
- **Control flow (if-else, switch, loops)**
- **Functions (declaration, expression, arrow functions)**
- **Arrays and array methods**

– DOM Manipulation

- Understanding the Document Object Model (DOM)
- Selecting and manipulating DOM elements
- Handling events (click, keypress, submit, etc.)
- Modifying HTML and CSS through the DOM

– JavaScript Objects

- Creating objects using object literals
- Accessing and modifying object properties
- Methods and `this` keyword
- Prototype-based inheritance
- Object constructors and classes

– JavaScript Data Structures

- Arrays (indexing, methods, iterating)
- Objects (key-value pairs, nested objects)
- Sets and Maps
- Working with JSON (parsing and stringifying)

– Asynchronous JavaScript

- Callbacks and the event loop
- Promises and async/await
- Handling asynchronous operations (AJAX, fetch API)
- Timeouts and intervals

– ES6 and Beyond

- Introduction to ECMAScript 6 (ES6)
- Let, const, and block-scoped variables
- Arrow functions and lexical `this`
- Template literals and string interpolation
- Destructuring, spread, and rest operators

– Functions and Functional Programming

- First-class functions and higher-order functions
- Callback functions and function composition
- Functional programming concepts (map, filter, reduce)
- Closures and module pattern

– DOM Events and Browser APIs

- **Event handling (click, keypress, submit, etc.)**
- **Event propagation and delegation**
- **Timers (setTimeout, setInterval)**
- **Working with the Fetch API and XMLHttpRequest**

– JavaScript Tooling and Ecosystem

- **Package managers (npm, yarn)**
- **Module bundlers (Webpack, Rollup, Browserify)**
- **Transpilers and polyfills (Babel)**
- **Linters and code formatters (ESLint, Prettier)**
- **Testing frameworks (Jest, Mocha, Chai)**

– Advanced JavaScript Concepts

- **Prototypal inheritance and the prototype chain**
- **Scope and closures**
- **Asynchronous programming patterns**
- **Design patterns (constructor, module, singleton, etc.)**
- **Performance optimization techniques**

4 Bootstrap

– Introduction to Bootstrap

- **What is Bootstrap?**
- **Advantages of using Bootstrap**
- **Bootstrap versions and installation**

– Bootstrap Grid System

- **Grid concepts and layout**
- **Responsive grid tiers**
- **Grid examples and practices**

– Bootstrap Content

- **Typography**
- **Tables**
- **Figures and images**
- **Utilities (spacing, text, etc.)**

– Bootstrap Components

- **Buttons**
- **Navigation (navbars, dropdowns, etc.)**
- **Forms and inputs**
- **Alerts, badges, and progress bars**
- **Cards and modals**
- **Carousels and slideshows**

– Building Responsive Designs with Bootstrap

- **Responsive utilities**
- **Responsive breakpoints**
- **Responsive examples and practices**

– Customizing Bootstrap with Sass

- **Introduction to Sass**
- **Compiling Sass files**
- **Overriding Bootstrap variables**
- **Creating custom styles and components**

5 Tailwind CSS

– Introduction to Tailwind CSS

- **What is Tailwind CSS?**
- **The utility-first approach**
- **Benefits of using Tailwind CSS**
- **Setting up Tailwind CSS in a project**

– Tailwind CSS Fundamentals

- **Utility classes**

- **Responsive utilities**
- **Hover, focus, and state utilities**
- **Composition and combining utilities**

– Layout and Flexbox

- **The Tailwind CSS layout system**
- **Flexbox utilities**
- **Alignment and justification**
- **Responsive layouts**

– Grid and Box Alignment

- **Grid utilities**
- **Box alignment utilities**
- **Spacing utilities**

– Typography and Backgrounds

- **Text styling utilities**
- **Font families and sizes**
- **Background utilities**
- **Opacity and shadows**

– Borders, Effects, and Transitions

- **Border utilities**
- **Filter and blend mode utilities**
- **Transform utilities**
- **Transition utilities**

– Forms and Interactive Elements

- **Form input styling**
- **Form validation utilities**
- **Interactivity utilities**

– Customizing Tailwind CSS

- **Configuration files**
- **Defining custom utilities**
- **Extending and overriding default styles**

- **Working with plugins**

– Integrating Tailwind CSS

- **Using Tailwind CSS with React/ Vue.js/ Angular**
- **Combining Tailwind CSS with other CSS frameworks**
- **Integrating Tailwind CSS with build tools and frameworks**

6 React.js

– Getting Started with React

- **Overview of React and its key features**
- **Setting up a React development environment**
- **Creating a basic React application**

– React Fundamentals

- **React components (functional and class-based)**
- **JSX syntax and expressions**
- **Props and state management**
- **Lifecycle methods**

– React Components

- **Composing components**
- **Styling components with CSS**
- **Handling user input and events**
- **Conditional rendering**

– React State Management

- **State manipulation and updates**
- **Lifting state up and down the component tree**
- **Handling asynchronous state changes**
- **Introduction to React Hooks**

– React Hooks

- **useState, useEffect, useContext**
- **Custom hooks and their use cases**
- **Hooks for data fetching and side effects**

React Router

- **Setting up client-side routing**
- **Handling dynamic routes and parameters**
- **Implementing navigation and link management**

React Forms and Validation

- **Controlled and uncontrolled form components**
- **Handling form submission and validation**
- **Integration with third-party form libraries**

React and API Integration

- **Fetching data from APIs**
- **Handling loading and error states**
- **Optimizing network requests**

State Management with Redux

- **Introduction to Redux and its core concepts**
- **Setting up a Redux store and connecting components**
- **Reducers, actions, and middleware**

Advanced React Concepts

- **Higher-Order Components (HOCs)**
- **Render props pattern**
- **Code splitting and lazy loading**
- **Performance optimization techniques**

React Testing

- **Unit testing with libraries like Jest and Enzyme**
- **Integration testing with tools like React Testing Library**
- **End-to-end testing with Cypress**

– React Ecosystem and Tooling

- **Popular React libraries and frameworks (Material-UI, Ant Design, NextJS, etc.)**
- **Build tools (Webpack, Rollup, Parcel)**
- **Deployment strategies and hosting options**

– React for Mobile Development

- **Introduction to React Native**
- **Building native mobile apps with React**
- **Platform-specific considerations and APIs**

– Real-World React Projects

- **Hands-on development of complex React applications**
- **Utilizing best practices and design patterns**
- **Incorporating state management, routing, and API integrations**

Any One Database (MongoDB or MySQL)

7 MongoDB

– Introduction to NoSQL and MongoDB

- **Overview of NoSQL databases**
- **MongoDB's advantages and use cases**
- **MongoDB architecture and components**

– MongoDB Installation and Setup

- **Installing MongoDB on different platforms**
- **MongoDB shell and GUI tools**
- **Connecting to a MongoDB instance**

– MongoDB Data Model

- **JSON and BSON data formats**
- **Documents and collections**
- **Schema design considerations**

– CRUD Operations

- **Inserting, querying, updating, and deleting documents**
- **Query operators and expressions**
- **Projections and embedded documents**

– Indexing and Performance

- **Understanding indexes in MongoDB**
- **Creating and managing indexes**
- **Query optimization techniques**

– Aggregation Framework

- **Introduction to the Aggregation Pipeline**
- **Aggregation stages and operators**
- **Data transformations and analytics**

– Data Modeling and Schema Design

- **Embedded vs. referenced data models**
- **Schema validation and data governance**
- **Case studies and best practices**

– Replication and Sharding

- **MongoDB replication architecture**
- **Setting up and managing replica sets**
- **Sharding for horizontal scaling**

– MongoDB Integration

- **Integrating MongoDB with programming languages (e.g., Python, JavaScript, Java)**
- **MongoDB drivers and ODM (Object-Document Mapping) libraries**
- **Using MongoDB in web applications**

– Administration and Security

- **User management and authentication**
- **Backup and restore strategies**
- **Performance monitoring and optimization**

– Advanced Topics (Optional)

- **MongoDB Atlas (cloud-hosted MongoDB service)**
- **Change streams and oplog**
- **MongoDB Stitch (serverless platform)**

8 MySQL

– Introduction to Databases and SQL

- **Overview of database management systems**
- **Relational database concepts**
- **Introduction to SQL (Structured Query Language)**

– MySQL Installation and Setup

- **Installing MySQL on different platforms**
- **MySQL command-line client and GUI tools**
- **Connecting to a MySQL server**

– SQL Basics

- **Creating and managing databases**
- **Data types and table definition**
- **Inserting, querying, updating, and deleting data**

– SQL Queries and Operators

- **SELECT statements and filtering data**
- **Joins (inner, outer, cross, and self)**
- **Aggregate functions and grouping**
- **Subqueries and derived tables**

– Database Design and Normalization

- **Entity-Relationship (ER) modeling**
- **Normalization forms and principles**
- **Indexing and performance optimization**

– Advanced SQL Topics

- Views and materialized views
- Stored procedures and functions
- Triggers and events
- Transactions and concurrency control

– MySQL Administration

- User management and security
- Backup and restore strategies
- Monitoring and performance tuning

– MySQL Programming

- Integrating MySQL with programming languages (e.g., Python, Java, PHP)
- MySQL drivers and connectors
- Object-Relational Mapping (ORM) tools

– Advanced MySQL Topics (Optional)

- Replication and high availability
- Partitioning and sharding
- MySQL in cloud environments

9 Python Language

– Introduction to Python

- What is Python? History and features of the language
- Installing Python and setting up the development environment
- Basic Python syntax and structure
- Python data types: integers, floats, strings, booleans, etc.
- Variables and assignment

– Python Operators and Expressions

- Arithmetic operators
- Comparison operators
- Logical operators
- Bitwise operators

- **Operator precedence**
- **Type conversion and casting**

– Python Collections (Data Structures)

- **Lists**
- **Tuples**
- **Dictionaries**
- **Sets**
- **List/dictionary/set comprehensions**

– Python Control Structures

- **If-else statements**
- **Nested if-else**
- **Loops (for, while)**
- **Break and continue statements**
- **Match-case statements**

– Python Functions

- **Defining and calling functions**
- **Function arguments and parameters**
- **Return statements**
- **Scope of variables**
- **Anonymous (lambda) functions**
- **Recursive functions**

– Python Exception Handling

- **Syntax errors and exceptions**
- **try-except blocks**
- **Raising and handling custom exceptions**
- **Context managers**

– Python Modules and Packages

- **Importing modules**
- **Creating and using custom modules**
- **Python standard library**

- **Third-party packages and libraries**
- **Virtual environments**

– Python File Handling

- **Opening and closing files**
- **Reading from and writing to files**
- **File modes (text, binary)**
- **Handling file paths**
- **Directories and folders**

– Object-Oriented Programming in Python

- **Classes and objects**
- **Attributes and methods**
- **Inheritance and polymorphism**
- **Encapsulation and information hiding**
- **Magic/dunder methods**

– Python Functional Programming

- **Functional programming concepts**
- **Lambda functions**
- **Map, filter, and reduce**
- **Generators and generator expressions**

– Python Testing and Debugging

- **Unit testing with unittest and pytest**
- **Test-driven development**
- **Debugging techniques and tools**

10 Django framework

– Introduction to Django

- **What is Django?**
- **Features of Django**
- **Django architecture (Model-View-Template)**

- **Setting up the development environment**

– Django Basics

- **Creating a new Django project**
- **Understanding the project structure**
- **Creating a new app**
- **URL routing and views**
- **Templates and template inheritance**
- **Static files handling**

– Models and Databases

- **Introduction to models**
- **Defining models**
- **Database migrations**
- **Model relationships (one-to-one, one-to-many, many-to-many)**
- **Querying data using the ORM**

– Forms and User Input

- **Django form handling**
- **Form validation**
- **Model forms**
- **CRUD operations using forms**

– User Authentication

- **Authentication system in Django**
- **User registration**
- **Login and logout**
- **Password management**
- **Permission and authorization**

– Advanced Django Topics

- **Class-based views**
- **Django admin interface**
- **Django signals**
- **Caching in Django**
- **Django middleware**
- **Deployment and scaling**

– Third-Party Packages

- **Introduction to Django packages**
- **Installing and using packages**
- **Popular packages (Django REST Framework, Django Channels, Django CMS, etc.)**

– Testing and Debugging

- **Writing unit tests in Django**
- **Test-driven development**
- **Debugging techniques**

– Django RESTful APIs

- **Introduction to RESTful APIs**
- **Django REST Framework**
- **Serializers**
- **Views and viewsets**
- **Authentication and permissions**

– Django and Front-end

- **Integrating Django with front-end frameworks (React/Angular/Vue.js)**
- **Django templates vs. front-end templates**
- **AJAX and Django**

– Other Important Topics

- **Django security best practices**
- **Performance optimization**

11 Git and GitHub

– Introduction to Version Control

- **Understanding version control systems**
- **Benefits of using Git and GitHub**
- **Setting up Git and GitHub accounts**

– Git Basics

- **Git repositories and workflow**
- **Initializing a Git repository**
- **Staging and committing changes**
- **Viewing commit history**

– Branching and Merging

- **Understanding branches in Git**
- **Creating, switching, and merging branches**
- **Resolving merge conflicts**

– Remote Repositories

- **Working with remote repositories on GitHub**
- **Pushing and pulling code**
- **Cloning existing repositories**

– Collaborative Development

- **Contributing to open-source projects on GitHub**
- **Creating and managing pull requests**
- **Code reviews and discussions**

12 Deployment and DevOps

- **Introduction to cloud platforms (AWS, Heroku, etc.)**
- **Containerization with Docker**
- **Continuous Integration and Continuous Deployment (CI/CD)**
- **Monitoring and logging**

Final Project: Students will work on a full-stack project incorporating the concepts learned throughout the course. This project will include both front-end and back-end components, database integration, user authentication, and deployment.

This syllabus provides a structured approach to learning full-stack development using Python, covering both front-end and back-end technologies along with essential topics such as databases, version control, deployment, and security.

SOFTCRAYONS®

PLACEMENT COMPANIES

SOFTCRAYONS®

Testimonials of Students

Sukhpreet Kaur
2 reviews

★★★★★ 2 weeks ago NEW

Hlo mam I'm Sukhpreet your Softcrayons tarining institute in student my training is digital marketing course 😄 I'm very becoz I'm digital marketing beginner but my experience to much becoz my trainer is very intelligent and supportive and nature is very friendly 😊

Manish Malik
1 review

★★★★☆ 2 weeks ago NEW

I'm new student in softcrayons my starting classes all gud my softcrayons experience to much better becoz my trainer is very experienced

Aman Bhardwaj
2 reviews

★★★★★ 5 months ago

I am Aman Bhardwaj, Recently I completed a Digital Marketing course from Softcrayons. After completing my course I got a placement at SNVA Ventures with a good salary package. If you want to do a course and boost your career in the Digital Marketing field. I will recommend you visit Softcrayons.
If I talk about the environment and faculty then Softcrayons have a very good and friendly environment and their faculty is highly experienced in the Digital Marketing field. Specifically, Yashvant sir is one of the best trainer and they have great experience in the Digital Marketing field.
Thank you Softcrayons and all staff who helped me boost my career in the Digital Marketing field.

Shivam Sharma
1 review

★★★★★ 2 months ago

I got the chance to study with the best teacher and they provided me a good career guidance. a veryb great place to learn programming and start your career.

Lalita Tiwari
1 review

★★★★★ 5 months ago

I heared about softcrayons through friends and I enroll myself here, and done my course. I suggest you all to join softcrayons. Hope you do great.

Tanish Chandrawal
5 reviews

★★★★★ 5 months ago

It is good institute, practical oriented practice is very good. This institute is very useful for graduate students to make carrier in IT. 100% job guarantee is available for all students.Very good Institute for Cloud Computing like Azure, AWS, GCP.

Aman Vishwakarma
2 reviews

★★★★★ 2 months ago

Hie guys I'm aman softcrayrons institute students for AutoCAD.. softcrayrons institute is best training institute sarfaraz sir is best teacher for softcrayrons. And best institute softcrayrons

SOFTCRAYONS®

Pearson

Skills for English SELT

kryterion by DRAKE INTERNATIONAL

GHAZIABAD

693, Sector 14-A, Vasundhara,
Ghaziabad, UP (201012)

NOIDA

B-132, Sector 2, Near Sector 15
Metro Station, Noida UP (201301)

info@softcrayons.com

+91 854 501 2345

www.softcrayons.com