

*Skills  
Today  
A Better  
Tomorrow!*

SAME  
LEARNER  
BIGGER  
OPPORTUNITIES

PROFESSIONAL PROGRAM

# FULL STACK WEB DEVELOPMENT

Learn. Build. Implement. From Code to Career!



## Why Choose Us?



### Practical Learning

Hands-on skills  
for real-world  
use



### Experienced Trainers

Learn from  
industry experts



### Career Focused

Build skills  
to get better  
opportunities



### Supportive Environment

A place to  
learn, practice  
and grow



*Same Learner  
Bigger Opportunities!*

*GOOD  
CODE  
BRIGHTER  
FUTURE*

# 1 Course Overview

This 6-month intermediate-level syllabus is designed to build practical skills for developing modern, database-driven web applications. The course covers frontend development, Python-based backend development, REST APIs, MySQL database management through phpMyAdmin, authentication, deployment, testing, security, Git/GitHub, and a final industry-style project.

## 2 Course Duration & Structure



### Duration

6 Months / 24 Weeks



### Level

Intermediate



### Recommended Study

15–20 hours/week



### Primary Stack

HTML, CSS, JavaScript,  
JavaScript, Python,  
Django/FastAPI,  
MySQL, Git/GitHub

## 5 24-Week / 6-Month Detailed Syllabus

### Month 1

### Frontend Foundations & Advanced HTML/CSS

#### Week 1

#### Web Fundamentals & HTML5



- How the web works: browser, server, HTTP/HTTPS, domain and hosting basics
- HTML document structure, semantic HTML, headings, paragraphs, links and images
- Lists, tables, forms, input types, validation and accessibility basics
- HTML5 semantic elements and SEO-friendly structure
- **Mini Project:** Personal Profile / Portfolio Page

## Week 2

### CSS3 Fundamentals



- Selectors, specificity, cascade and inheritance
- Box model, display, positioning, overflow and z-index
- Colors, typography, backgrounds, borders and shadows
- Flexbox fundamentals and practical layouts
- Responsive design principles and media queries

## Week 3

### Advanced CSS & UI Design



- CSS Grid and responsive page structures
- Transitions, transforms and animations
- Reusable components, CSS variables and maintainable CSS
- Bootstrap or Tailwind CSS fundamentals
- Responsive navigation, cards, forms, modals and dashboards

## Month 1

## Frontend Foundations & Advanced HTML/CSS

### Week 4

### Frontend Project & DevTools



- Chrome DevTools: Elements, Console, Network and responsive testing
- Form design and client-side validation
- Accessibility and cross-browser considerations
- Frontend project planning and folder structure
- **Project:** Responsive Admin Dashboard / Business Website

## Month 2

## JavaScript & JavaScript

### Week 5

### JavaScript Fundamentals



- Variables, data types, operators and control flow
- Functions, scope, arrays and objects
- Loops, array methods and object methods
- Destructuring, spread/rest and template literals
- ES6+ fundamentals

## Week 6

## DOM, Events &amp; Browser APIs



- DOM selection and manipulation
- Events and event delegation
- Form handling and validation
- LocalStorage and SessionStorage
- JSON, Fetch API and asynchronous JavaScript
- **Mini Project:** Expense Tracker / To-Do Application

## Week 7

## JavaScript Fundamentals



- JavaScript concepts, components and JSX
- Props, state and component lifecycle concepts
- Events, forms and conditional rendering
- Lists, keys and reusable components
- JavaScript project structure and best practices

## Month 2

## JavaScript & JavaScript

### Week 8

### JavaScript Intermediate



```
import { useState, useEffect }
from 'react';

function Dashboard() {
  const [data, setData] =
    useState([]);
  useEffect(() => {
    // Fetch data
  }, []);
  return <div>Dashboard</div>;
}
```



Routing



API  
Integration



State  
Management



- Hooks: useState, useEffect, useRef and custom hooks
- JavaScript routing concepts and protected routes
- API integration with Fetch/Axios
- Loading, error and empty states
- Context API and basic state management
- **Project:** Student Management Frontend / Product Dashboard

## Month 3

## Python Programming & Backend Foundations

### Week 9

### Python Fundamentals



```
# Python Basics
name = "Student"
age = 18

if age >= 18:
    print("Adult")
else:
    print("Minor")
```



Variables



Modules



File Operations

- Python syntax, variables, data types and operators
- Conditions, loops, functions and modules
- Lists, tuples, sets and dictionaries
- Exception handling and file operations
- Virtual environments, pip and project structure

## Week 10

### Object-Oriented Python



```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def display(self):
        print(f"Name: {self.name}")
        print(f"Age: {self.age}")
```



Classes



Objects



Inheritance

- Classes, objects, constructors and methods
- Encapsulation, inheritance and polymorphism
- Modules, packages and reusable code
- Decorators and useful Python standard-library concepts
- Debugging and clean-code practices

## Week 11

### Django Fundamentals

django

```
# Django Views
from django.shortcuts import render

def home(request):
    return render(request,
        'home.html')
```



Project Structure



URL Routing



Models



Templates

- Django architecture and project/app structure
- URL routing, views and templates
- Models and Django ORM
- Migrations and database configuration
- Admin panel and basic CRUD

## Week 12

### Backend Development



```
# Form Handling
from django import forms

class ProductForm(forms.ModelForm):
    class Meta:
        model = Product
        fields = ['name', 'price']
```



Forms & Validation



Sessions & Cookies



File Uploads



Project Organization

- Forms, validation and error handling
- Sessions, cookies and messages
- Static/media files and file uploads
- Reusable services/helpers and project organization
- **Mini Project:** Blog / Inventory CRUD Application

## Week 13

## MySQL &amp; Database Design



- Relational database concepts and normalization
- Tables, primary keys, foreign keys and constraints
- CREATE, ALTER, INSERT, UPDATE, DELETE and SELECT
- WHERE, ORDER BY, GROUP BY, HAVING and aggregate functions
- Database design using phpMyAdmin

## Week 14

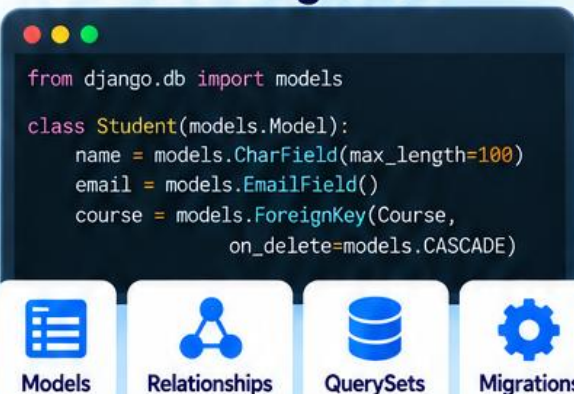
## Advanced SQL



- JOINS: INNER, LEFT, RIGHT and self joins
- Subqueries and correlated queries
- Indexes and query performance basics
- Transactions and ACID concepts
- Views and stored procedures overview
- Backup/import/export through phpMyAdmin

## Week 15

## Django ORM &amp; Database Integration



- Models, relationships and migrations
- One-to-one, one-to-many and many-to-many relationships
- QuerySets, filtering, annotations and aggregations
- Transactions and database optimization
- Practical database-driven modules

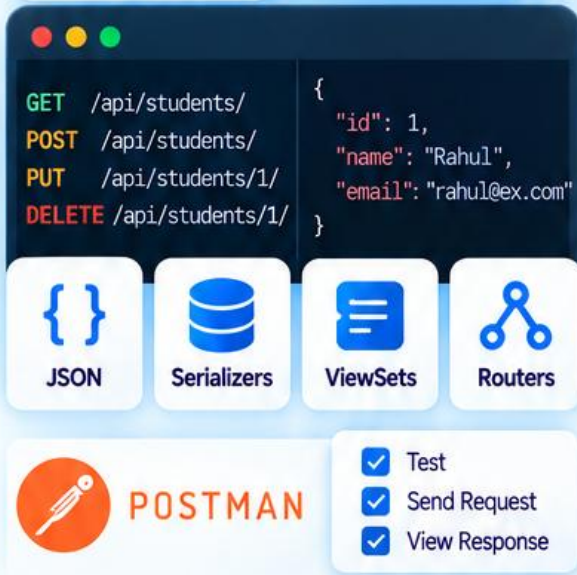
## Month 4

## MySQL, phpMyAdmin & REST API Development

### Week 16

### REST APIs

django  
**REST**  
framework



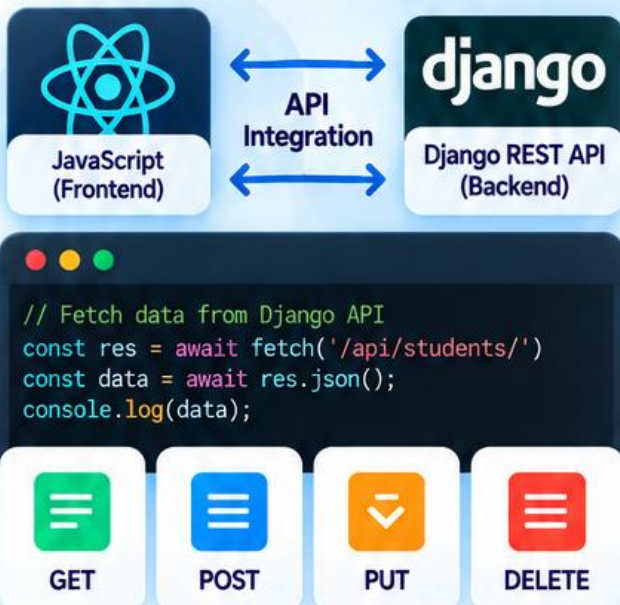
- REST architecture and HTTP methods
- JSON request/response handling
- Django REST Framework basics
- Serializers, API views/viewsets and routers
- Validation, pagination, filtering and search
- Testing APIs with Postman
- **Project:** Student/Employee REST API

## Month 5

## Full Stack Integration, Authentication & Security

### Week 17

### JavaScript + Python API Integration



- Connecting JavaScript with Django REST APIs
- GET, POST, PUT/PATCH and DELETE operations
- Authentication token handling
- Reusable API service layer
- Frontend forms connected to backend validation
- **Project:** Full CRUD application

# Month 5

## Full Stack Integration, Authentication & Security

### Week 18

#### Authentication & Authorization



- Registration, login and logout
- Password hashing and password reset concepts
- JWT/token-based authentication
- Role-based access control
- Protected API endpoints and protected frontend routes
- Admin/user permissions

### Week 19

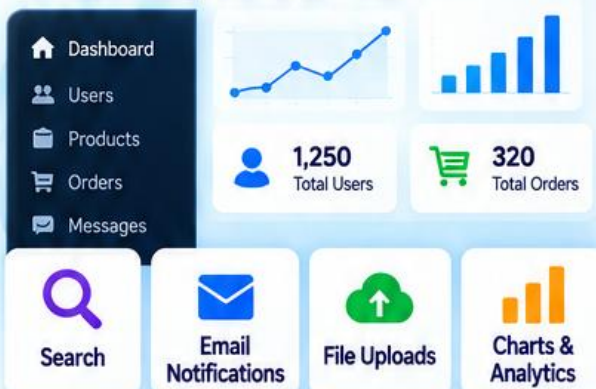
#### Security & Production Practices



- CSRF, XSS, SQL injection and secure coding concepts
- CORS and secure API configuration
- Input validation and server-side validation
- Environment variables and secret management
- File-upload security and access control
- Logging, error handling and basic monitoring

### Week 20

#### Advanced Full Stack Features



- Search, filtering, sorting and pagination
- Notifications and email integration concepts
- File/image upload and media management
- Dashboard statistics and charts
- Caching and performance basics
- Optional: WebSockets/realtime concepts

### Week 21

#### Git & GitHub

Version Control for Real Projects



Git



GitHub

```
$ git init
$ git add .
$ git commit -m "Initial commit"
$ git push origin main
```

*Code  
Collaborate  
Grow*



Commit



Branch



Merge



Push to  
GitHub

- Git fundamentals and repository setup
- Commit, branch, merge, rebase and conflict resolution
- GitHub repositories, README and project documentation
- Issue tracking and collaborative workflow
- Git best practices for professional projects

### Week 22

#### Testing & Deployment

Test. Deploy. Make it Live.



Testing



Postman



Linux



NGINX



aws  
Deployment

```
$ pytest
$ python manage.py test
$ gunicorn myproject.wsgi:application
$ sudo systemctl start nginx
```



Unit Testing



API Testing



Frontend  
Testing



Deployment

- Unit and integration testing concepts
- API testing with Postman
- Frontend testing overview
- Production settings and environment configuration
- Linux command-line basics
- Web server, reverse proxy and deployment concepts
- Deploying frontend, backend and MySQL database

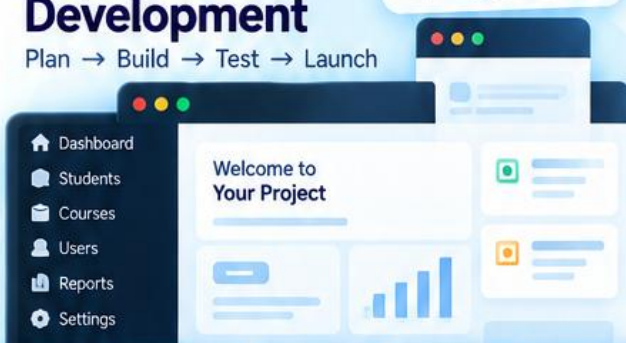
### Week 23

#### Capstone Development

Plan → Build → Test → Launch



Real Project  
Real Skills  
Real Future



Planning



Database  
Design



Development



Testing



Launch

- Requirement analysis and feature planning
- Database schema and ER-diagram preparation
- UI wireframe and frontend implementation
- Backend architecture and REST API development
- Authentication, authorization and CRUD modules
- Integration, testing and bug fixing



**Complete a Real-World Project**  
and be Industry Ready!

## Week 24

# Capstone Completion & Interview Preparation

Polish Your Skills. Present with Confidence.



- Performance optimization and security review
- Final testing and production deployment
- GitHub documentation and project presentation
- Code review and refactoring
- Technical interview questions and practical coding tasks
- Final project demonstration / viva



## 6. Practical Projects

*Learn by Building Real Projects!*

| Project                                                                                                  | Application                           | Technology                                    |                                                                                       |
|----------------------------------------------------------------------------------------------------------|---------------------------------------|-----------------------------------------------|---------------------------------------------------------------------------------------|
|  Project 1            | Responsive Personal Portfolio         | HTML, CSS, Bootstrap/Tailwind                 |  |
|  Project 2            | JavaScript Expense Tracker            | JavaScript, DOM, LocalStorage                 |  |
|  Project 3            | JavaScript Student/Product Dashboard  | JavaScript, Router, API concepts              |  |
|  Project 4            | Django CRUD Application               | Python, Django, MySQL                         |  |
|  Project 5            | REST API                              | Django REST Framework, Postman, MySQL         |  |
|  Project 6 - Capstone | Complete Full Stack Management System | JavaScript + Python/Django + MySQL + REST API |  |



**From Learning to Real-World Implementation**  
Build Projects. Gain Confidence. Become Job Ready!

*Your Project  
Your Future!*

## # 7. Recommended Capstone Project

Learn  
Build  
Implement

FROM CODE  
TO CAREER

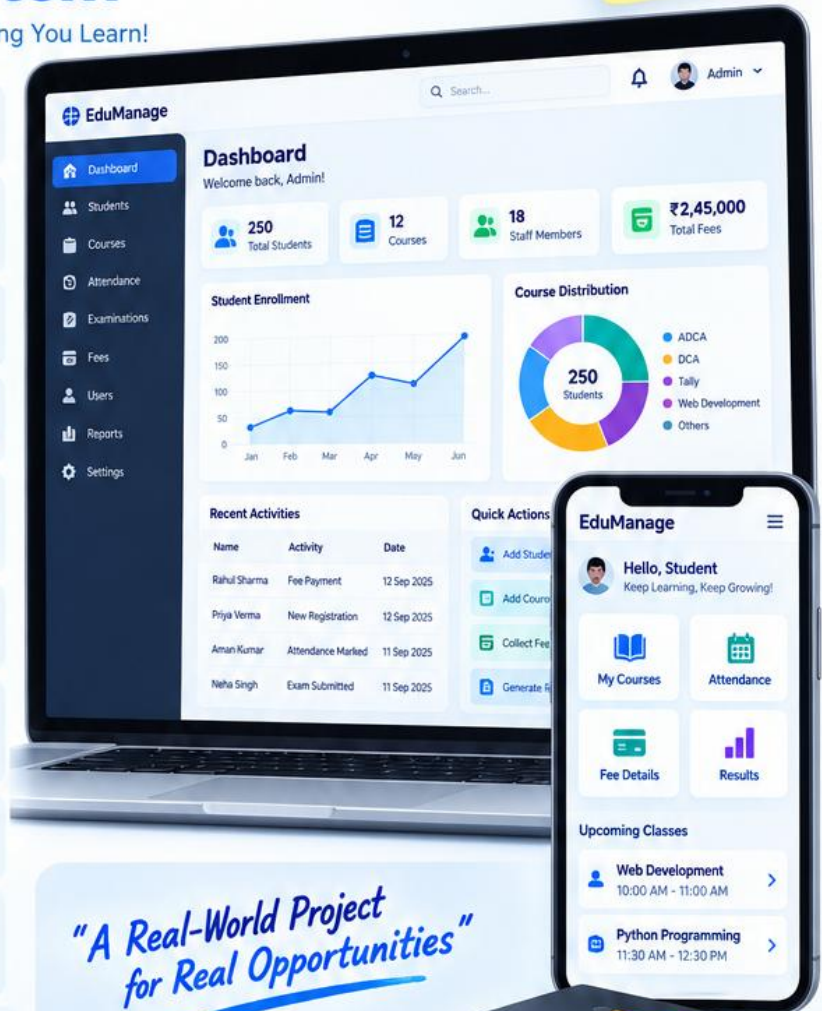
# Full Stack Student / Institute Management System

A Complete Real-World Project to Apply Everything You Learn!

"Manage Students.  
Streamline Operations.  
Build a Smarter Institute."

Your  
Capstone.  
Your Future.

- 01 User Authentication**  
With Admin, Staff and Student roles
- 02 Student Management**  
Student registration and profile management
- 03 Course & Subject Management**  
Manage courses, programs and subjects
- 04 Fee & Payment Records**  
Fee collection, payment history and reporting
- 05 Attendance & Examination**  
Attendance tracking and examination modules
- 06 Search, Filter & Pagination**  
Search, filtering, sorting and pagination
- 07 Dashboard & Analytics**  
Dashboard with summary statistics and charts
- 08 REST API**  
REST API for frontend communication
- 09 MySQL Database**  
MySQL relational database with properly designed relationships
- 10 Responsive Frontend**  
Modern and responsive JavaScript frontend
- 11 Python/Django Backend**  
Scalable and secure Python/Django backend
- 12 Deployment**  
Deployment and production configuration



"A Real-World Project  
for Real Opportunities"



### TECHNOLOGY STACK



HTML



CSS



JavaScript



React  
(JavaScript)



Python



Django



MySQL



DRF  
REST API



Git



GitHub



Postman



Deployment



Build Skills. Build Projects. Build Your Career!

Full Stack Today  
A Better Tomorrow



# 8.

# Weekly Study Pattern

A Balanced Plan for Steady Learning and Real Progress

Learn  
Code  
Practice  
Build  
Grow

Small  
Steps  
Big  
Results!

**Mon – Tue**



## Learn Concepts & Study Documentation

- Watch/attend sessions
- Read documentation
- Make notes
- Understand key concepts

Build a Strong Foundation

**Wed – Thu**



## Coding Exercises & Guided Implementation

- Practice coding
- Solve exercises
- Implement examples
- Take guidance when needed

Practice and Apply

**Friday**



## Module-level Practical Task

- Complete hands-on task for the module
- Implement features
- Test your code
- Get feedback

Turn Learning into Action

**Saturday**



## Project Development & Debugging

- Work on your project
- Add new features
- Debug issues
- Improve and refactor code

Build Something Real

**Sunday**



## Revision, GitHub Update & Weekly Assessment

- Revise what you learned
- Update code on GitHub
- Take weekly assessment
- Plan for next week

Reflect and Improve

# 9.

# Assessment Plan

Track Your Progress. Build Real Skills.

Consistent  
Assessment  
Better Future!



| Assessment                                                                                                             | Suggested Weight | Focus                                     | Details                                                                                                                                                                                    |
|------------------------------------------------------------------------------------------------------------------------|------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <b>Weekly Coding Tasks</b>         | <b>15%</b>       | Concept understanding and coding practice | <ul style="list-style-type: none"> <li>• Small coding exercises</li> <li>• Module-wise tasks</li> <li>• Code quality and logic</li> </ul>                                                  |
|  <b>Monthly Projects</b>            | <b>20%</b>       | Frontend/backend/database implementation  | <ul style="list-style-type: none"> <li>• Mini projects</li> <li>• Feature implementation</li> <li>• Practical understanding</li> </ul>                                                     |
|  <b>SQL &amp; API Tests</b>         | <b>15%</b>       | Database and REST API skills              | <ul style="list-style-type: none"> <li>• SQL queries and database design</li> <li>• REST API development</li> <li>• API testing (Postman)</li> </ul>                                       |
|  <b>Git/GitHub &amp; Deployment</b> | <b>10%</b>       | Professional development workflow         | <ul style="list-style-type: none"> <li>• Git commands and branching</li> <li>• GitHub repository and documentation</li> <li>• Deployment process</li> </ul>                                |
|  <b>Capstone Project</b>            | <b>30%</b>       | Complete full-stack application           | <ul style="list-style-type: none"> <li>• End-to-end project development</li> <li>• Implementation of all learned concepts</li> <li>• Code quality, functionality and deployment</li> </ul> |
|  <b>Final Viva/ Interview</b>       | <b>10%</b>       | Technical explanation and problem solving | <ul style="list-style-type: none"> <li>• Project demonstration</li> <li>• Conceptual and technical questions</li> <li>• Problem-solving skills</li> </ul>                                  |



**Assess. Improve. Grow.**

Every Task Brings You Closer to Your Dream Career!

Learn Today  
Lead Tomorrow!



# 10.

# Final Skill Set After 6 Months

From Learner to Job-Ready Full Stack Developer

Learn  
Build  
Practice  
Grow

01

## Build Responsive Websites & SPAs



HTML

CSS

JS



02

## Write Clean Code

Write clean intermediate-level JavaScript and Python.



JS



03

## Build Django Applications & REST APIs

Build Django applications and REST APIs.

django · API



04

## Design & Manage MySQL Databases

Design MySQL databases and manage them using phpMyAdmin.



05

## Perform CRUD & Complex SQL Queries

Perform CRUD operations and complex SQL queries.

SELECT  
INSERT  
UPDATE  
DELETE

06

## Connect Frontend with Backend APIs

Connect JavaScript frontend with Python backend APIs.



07

## Authentication & Role-Based Permissions

Implement authentication, authorization and role-based permissions.



Admin



Staff



Student

08

## Use Git/GitHub

Use Git/GitHub and professional project structures.



Git



GitHub



09

## Debug Applications

Debug applications using browser and backend tools.



Browser DevTools



Django Debug Toolbar



Logs &amp; Console

10

## Security, Testing & Deployment

Apply basic security, testing, optimization and deployment practices.



Security



Testing



Optimization



Deployment

11

## Build a Portfolio-Ready Full-Stack Project

Build and present a complete portfolio-ready full-stack project.

Real Projects  
Real Opportunities!Be Job-Ready  
in 6 Months!

Practical Skills



Real Projects



Better Career Opportunities

Skills Today.  
A Brighter Tomorrow.

Build. Apply. Grow.



## # 11. Suggested Prerequisites

Get Ready Before You Start

01



### Basic computer and internet usage

Comfortable with operating system, files/folders, and online tools.

02



### Basic programming logic is helpful but not mandatory

Logical thinking and problem-solving skills will be useful.

03



### Basic familiarity with HTML/CSS is recommended

Helps you get started faster with frontend development.

04



### A laptop/desktop capable of running required tools

Your system should be able to run VS Code, Python, MySQL and a modern browser.

- ✓ VS Code
- ✓ Python
- ✓ MySQL
- ✓ Modern Browser

Good Preparation Leads to Better Learning!

- ☐ Right Tools
- ☐ Positive Mindset
- ☐ Consistent Effort
- ☐ Bright Future!

Same Learner Bigger Opportunities!

Learn  
Practice  
Build  
Grow

CODE  
LEARN  
GROW  
REPEAT

Step by Step

## # 12. Important Note

Focus on Practice, Not Just Completion

Less Today A Better Tomorrow!

“

The syllabus is **intentionally practical**. The target is not only to complete technologies but to develop the ability to **design, code, debug, connect, test and deploy** a complete application.

Students should spend more time **writing code and building projects** than watching tutorials.

”



Practice More



Build Real Projects



Develop Real Skills



Become Job-Ready

Less Watching More Doing!

DISCIPLINE TODAY  
SUCCESS TOMORROW



PRACTICE TODAY  
**PROFESSIONAL TOMORROW**



Learn



Build



Grow

Your Skills  
Our Support  
A Brighter Future!

## OUR COMMITMENT



### Quality Education

Practical & industry-relevant learning



### Student Support

Guidance at every step



### Career Growth

Skills for real opportunities



### A Brighter Tomorrow

Because your success matters

*Skills  
Today  
A Better  
Tomorrow!*

*Code  
Learn  
Create  
Grow  
Together*

**BFS**  
COMPUTER INSTITUTE  
— Learn • Practice • Grow —

*"Good  
Skills  
Bright  
Futures."*



### Learn Anytime, Anywhere!

Download the BFS Learning App  
on Play Store



Video  
Classes



Study  
Material



Online  
Tests



Course  
Updates



Near Primary School,  
Ranipur Road, Mariahu, Jaunpur

9305652171, 7991332016

[www.bfsedu.org](http://www.bfsedu.org)

*Follow Us*



/bfscomputerinstitute