

## Practical 2: Career-Oriented Presentation

### Aim

To create a career presentation using slides, transitions, and animations.

### Objectives

- To design a multi-slide professional presentation
- To apply transitions and animations

### Materials Required

- PowerPoint or Google Slides

### Procedure

#### **Open a blank presentation**

Launch PowerPoint/Google Slides and select the option to create a new blank presentation.

This opens a fresh workspace where you will design your slides.

#### **Create a title slide**

Insert a title slide layout and add the presentation title along with your name or subtitle. Ensure the title is clear, readable, and visually centered on the slide.

#### **Add minimum 7 slides**

Use the “New Slide” option to insert at least seven additional slides with appropriate layouts.

Each slide should focus on a single topic or idea for clarity.

#### **Insert images, icons, and bullet points**

Add relevant images and icons to visually support your content.

Use bullet points to present information in a structured and easy-to-read format.

#### **Apply a theme**

Choose a professional theme from the design options available in the software.

The theme will automatically set consistent fonts, colors, and backgrounds.

### Add transitions and animations

Apply slide transitions for smooth movement between slides.

Add animations to text or images to enhance the presentation without overusing effects.



# Roadmap to Become a Web Developer in 2025

Your complete guide to mastering web development and launching a successful career in tech

## Why Become a Web Developer?



### High Demand

Web development jobs are booming globally with opportunities in startups, enterprises, and freelancing



### Competitive Pay

Full-stack developers earn excellent salaries thanks to their versatile, in-demand skill sets



### Build Impact

Create scalable applications that solve real problems and reach millions of users worldwide

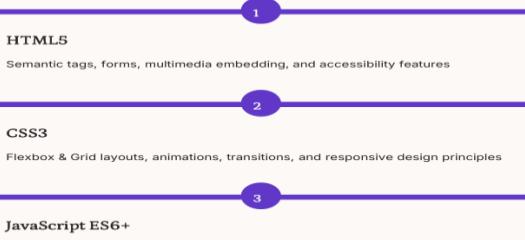


### Never Stop Learning

Constant innovation ensures exciting challenges and continuous growth throughout your career

## Step 1: Master the Frontend Basics

### Essential Technologies



**Practice makes perfect:** Build static, interactive web pages to solidify your foundation. Start with simple landing pages and gradually add interactivity.

## Step 2: Choose a Frontend Framework



### React.js

The most popular choice for dynamic UIs and single-page applications. Backed by Meta with a massive ecosystem.

- Component-based architecture
- State management with hooks
- Virtual DOM for performance



### Vue.js

Lightweight and beginner-friendly framework perfect for progressive integration into projects.



### Angular

Enterprise-ready framework with TypeScript, ideal for large-scale applications requiring structure.



**Build real projects:** Create portfolios, to-do apps, weather dashboards, or personal blogs to showcase your skills.

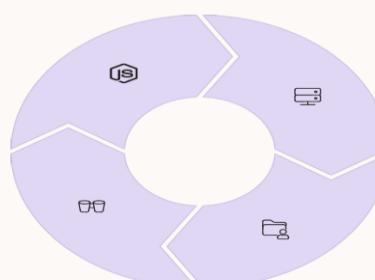
## Step 3: Dive into Backend Development

### Choose Your Language

Node.js, Python (Django/Flask), or Java (Spring Boot)

### Connect Frontend

Build APIs and integrate with your frontend applications



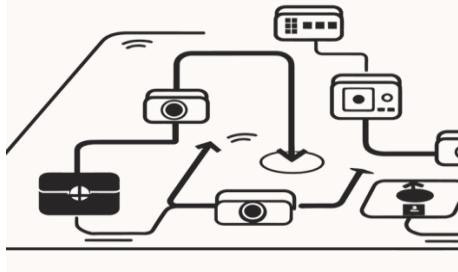
### Server Logic

RESTful APIs, authentication (JWT/OAuth), and routing

### Database Skills

SQL (PostgreSQL, MySQL) and NoSQL (MongoDB)

## Step 5: Advanced Tools & Deployment



### CI/CD Pipelines

Automated testing and deployment for faster, more reliable releases

### Cloud Platforms

Deploy on Vercel, Netlify, Heroku, AWS, or Google Cloud for global reach

### Docker Containers

Create consistent development and production environments

### Security & Performance

Monitor metrics, implement HTTPS, and follow security best practices

These advanced skills separate junior developers from professionals. Master them to stand out in the job market.

## Step 4: Version Control & Collaboration

01

### Learn Git Basics

Master commits, branches, merges, and resolving conflicts effectively

02

### GitHub/GitLab

Host your code, showcase projects, and collaborate with developers worldwide

03

### Pull Requests

Understand code reviews, collaboration workflows, and best practices

04

### Team Workflows

Essential for professional development environments and open source contributions



**Pro tip:** Commit early and often with clear, descriptive messages. Your future self will thank you!

## Step 6: Embrace Modern Trends & AI Tools

### Progressive Web Apps

Build web apps that feel native with offline capabilities and push notifications

### WebAssembly

Near-native performance for compute-intensive web applications

### Serverless Architecture

Scale automatically and pay only for what you use with cloud functions

### AI-Powered Tools

Boost productivity with GitHub Copilot and other intelligent coding assistants

*Stay ahead of the curve by learning cutting-edge technologies. The future of web development is here!*

## Step 7: Build Real Projects & Network



### Diverse Portfolio

E-commerce sites, blogs, dashboards, and SaaS applications



### Open Source

Contribute to projects and join hackathons for experience



### Join Communities

Discord, Stack Overflow, Reddit, and local meetups



### Keep Learning

Adapt to new technologies and industry trends



**Remember:** Your GitHub profile is your resume. Make it count with quality projects and consistent contributions!

# Your Journey Starts Now!

### Start Small, Build Consistently

Never stop learning. Every expert was once a beginner who didn't give up.

### Free Resources Await

Leverage freeCodeCamp, The Odin Project, roadmap.sh, and countless YouTube tutorials.

### The Web is Your Canvas

Create, innovate, and shape the future of the internet with your unique vision.

"The best time to start was yesterday. The second best time is now. Ready to code your future? Let's build the web together!"

[Start Learning](#)

