Web Development Guide

Web Development creates websites and web applications.

FRONTEND DEVELOPMENT:

1. HTML (HyperText Markup Language)
   - Structure of web pages
   - Tags: <html>, <head>, <body>, <div>, <p>, <a>
   - Semantic HTML5: <header>, <nav>, <article>, <footer>

   Basic Structure:
```
<!DOCTYPE html>
<html>
<head>
    <title>My Website</title>
</head>
<body>
    <h1>Welcome</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```
2. CSS (Cascading Style Sheets)
   - Styling and layout
   - Selectors: element, class, ID
   - Box Model: margin, border, padding, content
   - Flexbox: 1D layout
   - Grid: 2D layout
   - Responsive Design: Media queries

   Example:
```
.container {
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #f0f0f0;
}
```
3. JavaScript
   - Programming language for interactivity
   - DOM Manipulation
   - Event Handling
   - Asynchronous Programming (Promises, async/await)
   - ES6+ Features: Arrow functions, destructuring, spread operator

   Example:
```
document.getElementById('btn').addEventListener('click', () => {
    alert('Button clicked!');
});
```

FRONTEND FRAMEWORKS:

1. React (Facebook)
   - Component-based architecture
   - Virtual DOM for performance
   - JSX syntax
   - Hooks: useState, useEffect
2. Angular (Google)
   - Full-featured framework
   - TypeScript-based
   - Two-way data binding
   - Dependency injection
3. Vue.js
   - Progressive framework
   - Easy to learn
   - Reactive data binding
   - Single-file components

BACKEND DEVELOPMENT:

Server-side logic and database management.

1. Node.js (JavaScript)
   - Non-blocking I/O
   - npm package manager
   - Express.js framework

   Example:
   ```
   const express = require('express');
   const app = express();
   app.get('/', (req, res) => {
       res.send('Hello World!');
   });
   app.listen(3000);
   ```
2. Python (Django/Flask)
   - Django: Full-featured framework
   - Flask: Lightweight, flexible
   - Great for APIs
3. PHP
   - Server-side scripting
   - WordPress, Laravel
   - Easy to deploy
4. Java (Spring Boot)
   - Enterprise applications
   - Robust and scalable
   - Microservices architecture

DATABASES:

1. SQL Databases (Relational)
   - MySQL, PostgreSQL, SQLite

- Structured data with relationships
- ACID compliant

Example:

```
CREATE TABLE users (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    email VARCHAR(100)
);
SELECT * FROM users WHERE age > 18;
```

2. NoSQL Databases
   - MongoDB (Document)
   - Redis (Key-Value)
   - Cassandra (Wide-Column)
   - Flexible schema, horizontal scaling

RESTful APIs:

REST (Representational State Transfer)

HTTP Methods:

- GET: Retrieve data
- POST: Create new resource
- PUT: Update existing resource
- DELETE: Remove resource

Example API:

```
GET    /api/users       - Get all users
GET    /api/users/1     - Get user with ID 1
POST   /api/users       - Create new user
PUT    /api/users/1     - Update user 1
DELETE /api/users/1     - Delete user 1
```

WEB SECURITY:

1. HTTPS
   - Encrypted communication
   - SSL/TLS certificates
2. Authentication
   - JWT (JSON Web Tokens)
   - OAuth 2.0
   - Session-based auth
3. Common Attacks
   - SQL Injection: Sanitize inputs
   - XSS (Cross-Site Scripting): Escape output
   - CSRF (Cross-Site Request Forgery): Use tokens
   - DDoS: Rate limiting, firewalls

DEPLOYMENT:

1. Hosting Platforms
   - Heroku: Easy deployment
   - AWS: Scalable cloud services

  - Netlify/Vercel: Frontend hosting
  - DigitalOcean: VPS hosting

2. Containerization
   - Docker: Package app with dependencies
   - Kubernetes: Orchestrate containers
3. CI/CD
   - GitHub Actions
   - Jenkins
   - GitLab CI
   - Automated testing and deployment

WEB PERFORMANCE:

1. Optimization Techniques
   - Minification: Remove whitespace
   - Compression: Gzip files
   - Caching: Store frequently accessed data
   - CDN: Content Delivery Network
   - Lazy Loading: Load content as needed
2. Performance Metrics
   - Page Load Time
   - Time to First Byte (TTFB)
   - First Contentful Paint (FCP)
   - Lighthouse Score

PROGRESSIVE WEB APPS (PWA):

- Works offline
- Installable on mobile
- Push notifications
- Service workers
- Manifest file