WELCOME

TO

This is an Education Platform

We Provide PDF Notes for Pharmacy Students

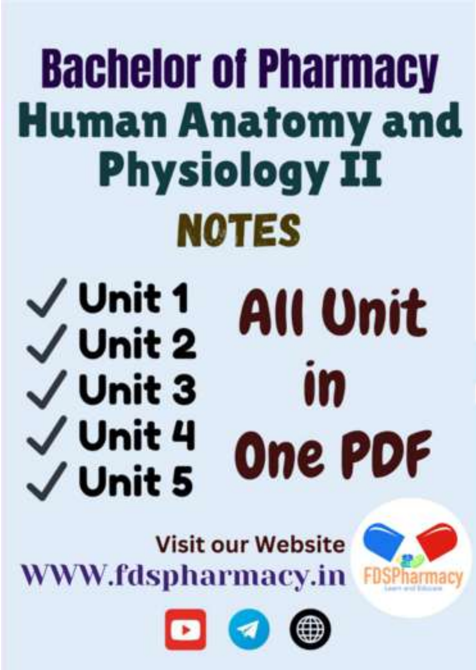
Web Site http://www.fdspharmacy.in/

You tube https://www.youtube.com/c/FDSpharmacy

Telegram https://t.me/Fdspharmacy

App https://play.google.com/store/apps/details?id=com.FDSPharmacyMedia.FDSPharmacy

E-mail fdspharmacyinfo@gmail.com

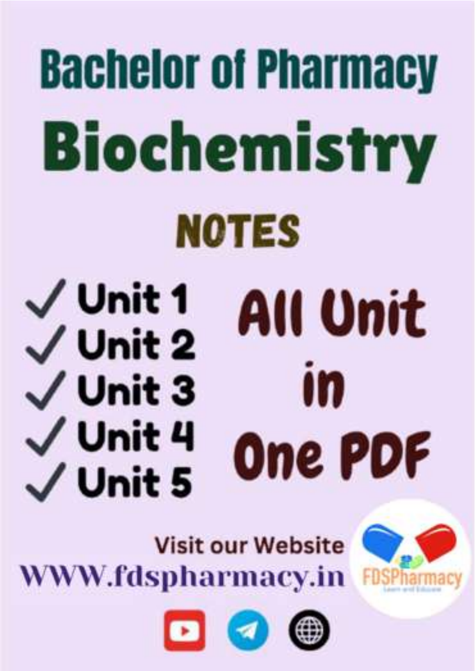

Bachelor of Pharmacy
Human Anatomy and Physiology II
NOTES
✓ Unit 1
✓ Unit 2
✓ Unit 3
✓ Unit 4
✓ Unit 5
All Unit in One PDF
Visit our Website
WWW.fdspharmacy.in
FDSPharmacy

Bachelor of Pharmacy
Pharmaceutical Organic Chemistry I
NOTES
✓ Unit 1
✓ Unit 2
✓ Unit 3
✓ Unit 4
✓ Unit 5
All Unit in One PDF
Visit our Website
WWW.fdspharmacy.in
FDSPharmacy

Bachelor of Pharmacy
Pathophysiology
NOTES
✓ Unit 1
✓ Unit 2
✓ Unit 3
✓ Unit 4
✓ Unit 5
All Unit in One PDF
Visit our Website
WWW.fdspharmacy.in
FDSPharmacy

Bachelor of Pharmacy
Environmental Sciences
NOTES
✓ Unit 1
✓ Unit 2
✓ Unit 3
✓ Unit 4
✓ Unit 5
All Unit in One PDF
Visit our Website
WWW.fdspharmacy.in
FDSPharmacy

Bachelor of Pharmacy
Computer Applications in Pharmacy
NOTES
✓ Unit 1
✓ Unit 2
✓ Unit 3
✓ Unit 4
✓ Unit 5
All Unit in One PDF
Visit our Website
WWW.fdspharmacy.in
FDSPharmacy


Bachelor of Pharmacy
Biochemistry
NOTES
✓ Unit 1
✓ Unit 2
✓ Unit 3
✓ Unit 4
✓ Unit 5
All Unit in One PDF
Visit our Website
WWW.fdspharmacy.in
FDSPharmacy

COMPUTER APPLICATIONS IN PHARMACY

UNIT 2

TOPIC :

- **Web technologies :** Introduction to HTML, XML, CSS and Programming languages, introduction to web servers and Server Products Introduction to databases, MYSQL, MSACCESS, Pharmacy Drug database

Introduction to Web Technologies

→ Web technologies refer to the tools and techniques used for communication between different types of devices over the internet. They form the foundation of the World Wide Web (WWW) and enable users to create, manage, and access websites and web applications.

What is the Web?

- The World Wide Web (WWW) is a system of interlinked hypertext documents accessed via the internet.
- Users access content using a web browser (e.g., Chrome, Firefox, Safari).
- Websites consist of web pages created using web technologies like HTML, CSS, JavaScript, etc.

Key Components of Web Technologies

1. HTML (HyperText Markup Language)

- Standard language for creating web pages.
- Uses **tags** (e.g., <h1>, <p>, <a>) to structure content like headings, paragraphs, links.
- HTML provides **structure**, not style.

2. CSS (Cascading Style Sheets)

- Used to **style** HTML elements.
- Controls layout, fonts, colors, and spacing.
- Example: h1 { color: blue; } makes heading text blue.

3. JavaScript

- A programming language used to add **interactivity** to websites.

- Enables features like dropdown menus, form validation, sliders, animations.
- Runs in the browser, making pages dynamic.

4. XML (eXtensible Markup Language)

- Used to store and transport data.
- Focuses on data structure, not display.
- Often used in web services and APIs.

Introduction to HTML (HyperText Markup Language)

- HTML stands for Hyper Text Markup Language.
- It is used to design web pages using markup tags.
- HTML combines:
  - Hypertext: Refers to the links between web pages.
  - Markup Language: Used to define the structure of content using tags.
- HTML is not a programming language; it is a markup language that annotates text for computer interpretation.
- Tags in HTML define how the content should be displayed on a web page.
- Most markup languages, including HTML, are human-readable.
- HTML enables browsers to manipulate and display text, images, links, audio, video, etc.

History and Versions of HTML

HTML Version	Year Released
HTML 1.0	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

- **HTML 2.0** was the first **standard version.**
- **HTML5** is the **latest version** with added support for video, audio, graphics, and semantic elements.

Structure of an HTML Page

HTML documents follow a specific structure that includes elements like <!DOCTYPE>, <html>, <head>, and <body>.

Basic HTML Structure Example

```
<!DOCTYPE html>

<html>

 <head>

  <title>Page Title</title>

 </head>

 <body>

  <h1>My First Heading</h1>

  <p>My first paragraph.</p>

 </body>

</html>
```

Explanation of Tags:

Tag	Description
<!DOCTYPE html>	Declares the HTML version (HTML5)
<html>	Root element of the HTML page
<head>	Contains metadata like title, CSS, JavaScript, etc.
<title>	Sets the browser tab title
<body>	Contains visible content like text, images, links, etc.

Tags allowed inside `<head>`

- <title>
- <style>
- <base>
- <noscript>
- <script>
- <meta>
- <link>

Features of HTML

- ✓ Easy to learn and implement
- ✓ Platform-independent
- ✓ Supports images, audio, video
- ✓ Allows hyperlinking (using <a> tag)
- ✓ Markup-based language
- ✓ Used to build the structure of web pages
- ✓ Compatible with all web browsers
- ✓ Can be integrated with CSS, JavaScript, etc.

Disadvantages of HTML

- ❖ Only static web pages can be created (no interactivity)
- ❖ Requires large amounts of code even for simple pages
- ❖ Poor security features (no in-built encryption or validation)

Introduction to XML (Extensible Markup Language)

- XML stands for Extensible Markup Language.
- It is a markup language similar to HTML but with a different purpose.
- Unlike HTML, XML tags are not predefined — users must define their own tags.
- XML is used to store and transport data, not display it.
- It is self-descriptive and often uses DTD (Document Type Definition) or XSD (XML Schema Definition) to define the document structure.

Key Features of XML

- ✓ XML is used to describe data.
- ✓ It stores data in plain text format, making it software- and hardware-independent.
- ✓ It allows users to create custom tags based on their needs.
- ✓ XML is extensible, meaning users can define their own structure and rules.

XML vs HTML

Feature	XML	HTML
Purpose	To describe data	To display data
Tag Definition	User-defined tags	Predefined tags
Structure	Must be well-formed and nested correctly	More lenient syntax
Data Focus	Focus on what the data is	Focus on how data looks
Customization	Highly customizable	Fixed tag set

- In **XML**, tags like <to>, <from> are invented by the user.
- In **HTML**, tags like <p>, <h1>, <table> are standardized.

Why Use XML?

- To store structured data in a plain text format.
- To share data between different systems/platforms.
- To transport data over the internet.
- To ensure data integrity and compatibility across software.

Applications of XML

- Used in office tools like Microsoft Word/Excel (internal file format).
- Used in web development and web services (like SOAP, RSS).
- Acts as a base language for many communication protocols.
- Ideal for data interchange between platforms and applications.
- Common in e-commerce, healthcare systems, and scientific data sharing.

XML Syntax Rules

- Every **opening tag** must have a **closing tag.**
  - Example: <name>Najim</name>
- Tags must be **properly nested.**
- XML is **case-sensitive.**
  - <Name> and <name> are considered different.
- The document must have a **single root element** that contains all other elements.

Types of Tags in XML

1. **Start tag** : <section>
2. **End tag** : </section>
3. **Empty tag** (self-closing) : <line-break />

XML Example: A Simple Reminder

```
<note>
  <to>Tom</to>
  <from>Sam</from>
  <heading>Reminder</heading>
  <body>Don’t forget me this weekend</body>
</note>
```

Explanation:

- <note> is the root element.
- Other elements like <to>, <from>, <heading>, and <body> are user-defined tags.
- The structure is logical, hierarchical, and easy to understand.

Advantages of XML

- ✓ **Text-based** format:
  - Takes up less storage space.
  - Can be transmitted easily over networks.
- ✓ **Media versatility**:
  - XML data can be displayed differently in **HTML**, **mobile apps**, **videos**, or **DVDs**.
- ✓ **Reusable and modular**:
  - Parts of an XML document can be **modularized** and reused in other documents or applications.
- ✓ **Platform-independent**:
  - Data can be shared across various systems (Windows, Mac, Linux).
- ✓ **Supports Unicode**:
  - Suitable for data in **any language**.

Introduction to CSS (Cascading Style Sheets)

- ➢ CSS stands for Cascading Style Sheets.
- ➢ It is a stylesheet language used to control the presentation and layout of web pages.
- ➢ CSS is used to apply styles (like color, font, spacing, layout) to HTML content.
- ➢ It allows separation of content (HTML) from presentation (CSS), making web design more efficient and organized.

Key Features of CSS

- **Style Reusability**: Write once, use across multiple web pages.
- **Separation of Concerns**: Keeps structure (HTML) separate from design (CSS).
- **Device Flexibility**: Enables different designs for different devices (e.g., screen, print, mobile, Braille).
- **Powerful Presentation Control**: More design options than HTML.
- **Efficient and Clean Coding**: Improves readability for both developers and search engines.

Benefits of Using CSS

- ✓ Saves Time:
  - A single CSS file can style multiple web pages.
- ✓ Easy Maintenance:
  - Change in CSS affects all associated pages automatically.
- ✓ Faster Page Loading:
  - Reduces code repetition; fewer HTML tags needed.
- ✓ Improved Accessibility:
  - Supports alternate formats like speech, print, or Braille.
- ✓ Offline Browsing:
  - Supports local caching for offline access to web apps.
- ✓ SEO-Friendly:

- Clean separation of style and content improves search engine readability.

Uses of CSS

CSS can be used to control the following elements on a webpage:

- Text color
- Font type and size
- Paragraph spacing
- Column layout and size
- Background images
- Background colors
- Page layout and positioning
- Borders, margins, and padding
- Hover and transition effects
- Visibility and display of elements

Types of CSS

1. **Inline CSS**:
   - Written directly within an HTML tag.
   - Example:

```
<p style="color:blue;">This is blue text</p>
```

2. **Internal CSS**:

- Written inside <style> tag in the <head> section of an HTML document.
- Example:

```
<style>

  p { color: red; }

</style>
```

3. **External CSS**:

- Written in a separate .css file and linked to the HTML file.
- Example:

```
<link rel="stylesheet" href="style.css">
```

Example: CSS for Center Alignment and Color Change

```
<!DOCTYPE html>

<html>

<head>

  <style>

    .center {

      text-align: center;

      color: red;

    }

  </style>

</head>

<body>

  <h1 class="center">Red and center-aligned heading</h1>

  <p class="center">Red and center-aligned paragraph</p>

</body>

</html>
```

Notes:

- The **class selector** .center applies the style to both heading and paragraph.
- text-align: center; centers the text.
- color: red; changes text color to red.

Advantages of CSS

- ✓ Controls layout of multiple pages from one file.
- ✓ Reduces repetition in HTML coding.
- ✓ Easy to maintain and update styles.
- ✓ Offers advanced styling options like animations, transitions, gradients.
- ✓ Enhances user experience across different devices and browsers.

Introduction to Programming Languages

- A programming language is a formal set of instructions used to communicate with a computer and create executable software. It allows the programmer to write code that the computer can interpret and execute, either directly or after translation.
- Programming languages serve as a bridge between human thinking and machine-level operations, allowing developers to implement logic, solve problems, and automate tasks.

Characteristics of Programming Languages

1. **Machine-Independent**: Focused on problem-solving, not tied to specific hardware.
2. **Readable & Understandable**: Uses familiar words and math symbols, closer to natural language.
3. **Translatable**: Each instruction must be convertible to machine code.
4. **Structured Syntax**: Has defined grammar rules (syntax) and keywords.
5. **Portable**: Programs written in higher-level languages can be used on different systems.

Generations of Programming Language

1. First Generation Languages
2. Second Generation Languages
3. Third Generation Languages
4. Fourth Generation Languages
5. Fifth Generation Languages

1 ***First Generation:*** *Machine Language*

- Uses **binary code** (0s and 1s)
- Directly understood by computers
- **Advantages**:
  - Fast execution
  - No translation needed
- **Disadvantages**:
  - Difficult to write and debug
  - Requires memorization of binary codes

2 ***Second Generation:*** *Assembly Language*

- Uses **mnemonics** instead of binary
- Requires an **assembler** to convert code to machine language
- **Advantages**:
  - Easier than machine language
  - More readable and modifiable
- **Disadvantages**:
  - Machine-dependent
  - Still low-level and hardware-specific

3 ***Third Generation:*** *High-Level Languages (HLL)*

- Closer to **human language** (e.g., C, C++, Java, Python)
- Needs a **compiler or interpreter**
- **Advantages**:
  - Easy to learn and use
  - Portable and maintainable
- **Disadvantages**:
  - Slower than machine/assembly
  - Object code may be less optimized

4 ***Fourth Generation:*** *4GL (Application-Oriented)*

- More **abstract** and **user-friendly**
- Often **non-procedural**, focusing on **what** to do rather than **how**
- Examples: SQL, MATLAB, Oracle Reports
- **Advantages**:
  - Rapid application development
  - Easier GUI-based interfaces
  - Reduces programming effort
- **Disadvantages**:
  - Less control over hardware
  - Slower execution than 3GL in some cases

5 ***Fifth Generation:*** *AI & Natural Language-Based*

- Used in **Artificial Intelligence** and **Machine Learning**
- Examples: Prolog, LISP
- Features **parallel processing, logic-based programming,** and **natural language inputs**
- **Advantages**:
  - Supports AI development
  - Capable of learning and reasoning
- **Disadvantages**:
  - Complex and expensive to develop
  - Requires high-performance hardware

Purpose and Uses of Programming Languages

- To create software applications
- Automate tasks and processes
- Analyze data and solve mathematical problems
- Build websites, games, and mobile apps
- Enable communication between hardware and software

Comparison of Language Generations

Generation	Language Type	Complexity	Portability	Use Case
1st	Machine Language	Very High	No	Hardware-level tasks
2nd	Assembly Language	High	No	Embedded systems, device drivers
3rd	High-Level	Medium	Yes	General-purpose programming
4th	Application-Based	Low	Yes	Databases, Business applications
5th	AI-Based	Low	Yes	AI, NLP, Expert systems

Introduction to Web Servers and Server Products

- A Web Server is a software and hardware system that uses HTTP (HyperText Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. Its primary function is to store, process, and deliver web pages to users. This communication typically takes place through browsers like Chrome, Firefox, Safari, etc.
- When you enter a website URL in your browser, the browser sends a request to the server where the website is hosted. The web server processes the request and sends back the appropriate web page files (HTML, CSS, images, etc.).

Functions of a Web Server

- Handles client (browser) requests
- Delivers requested content (HTML, CSS, JavaScript)
- Manages server-side scripts (like PHP, ASP.NET)
- Ensures security with SSL/TLS encryption
- Logs access and error information
- Provides database connectivity
- Supports authentication and session management

Types of Web Servers

Web Server	Description
Apache HTTP Server	Open-source, widely used server maintained by the Apache Software Foundation.
Nginx	Known for high performance, low resource usage, and ability to serve many concurrent connections.
Microsoft IIS (Internet Information Services)	A web server developed by Microsoft, tightly integrated with Windows OS.
LiteSpeed	Commercial web server known for its speed, security, and compatibility with Apache configurations.
Tomcat	A server developed by Apache to run Java Servlets and JavaServer Pages (JSP). Not a full-featured web server but often used alongside others.

Popular Web Server Products

Server Product	Description	Platform
Apache HTTP Server	Most popular open-source web server. Flexible and robust with many modules.	Cross-platform
Nginx	Lightweight and high-performing server. Ideal for handling a large number of simultaneous connections.	Cross-platform
Microsoft IIS	Integrated with Windows. Supports ASP.NET and has a GUI for configuration.	Windows
LiteSpeed Web Server	High-performance alternative to Apache with built-in caching and security features.	Cross-platform
XAMPP/WAMP/MAMP/LAMP	Bundles that include Apache server, MySQL database, PHP, and other tools for local development.	Platform-specific
Node.js	JavaScript-based environment that allows you to create your own web server using frameworks like Express.js.	Cross-platform

Benefits of Using a Web Server

- ✓ Efficient content delivery
- ✓ Reliable connection handling
- ✓ Security via HTTPS
- ✓ Load balancing and scalability
- ✓ Centralized control over website configuration

Introduction to Databases

➢ A Database is a systematic collection of data that is organized in such a way that it can be easily accessed, managed, and updated. It allows users to efficiently store, retrieve, and manipulate data using a software application.

For example, in a **pharmacy**, a database might store information about:

- Drugs and their compositions
- Patients and their prescriptions
- Inventory of medicines
- Sales and billing details

Database Management System (DBMS)

→ A Database Management System (DBMS) is software that helps you interact with databases. It allows you to insert, update, delete, and query the data.

Examples of DBMS:

- MySQL
- MS Access
- Oracle
- SQL Server
- SQLite

Components of a Database

1. **Tables** – Basic structure to store data in rows and columns
2. **Fields (Columns)** – Attributes of data (e.g., drug name, expiry date)
3. **Records (Rows)** – Individual entries (e.g., each drug entry)
4. **Queries** – Request to retrieve data

5. **Forms** – User-friendly interface for data input
6. **Reports** – Summarized output of data

Types of Databases

Type	Description
Relational Database	Stores data in tables (rows and columns) with relationships. (e.g., MySQL, Oracle)
Hierarchical Database	Organizes data in a tree-like structure.
Network Database	Allows multiple relationships between data.
Object-Oriented Database	Stores data as objects (used in programming).
Distributed Database	Data is stored across multiple physical locations.

Advantages of Databases

- ✓ Easy and fast data retrieval
- ✓ Reduces data redundancy (no repetition)
- ✓ Maintains data integrity
- ✓ Enables multiple users to access data simultaneously
- ✓ Provides data security and access control
- ✓ Ensures backup and recovery of data
- ✓ Facilitates report generation

Use of Databases in Pharmacy

Application	Purpose
Drug Database	Stores drug details (name, composition, price, expiry)
Patient Records	Maintains health history and prescriptions
Inventory Management	Tracks stock of medicines and supplies
Billing System	Automated invoice generation
Clinical Data Management	Used in research and clinical trials

Introduction to MySQL (Structured Query Language – SQL)

- MySQL is a database software that enables users to create, retrieve, update, and manage data in a relational format using SQL.
- MySQL is an open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) to manage and operate databases. It is widely used in web applications, health care systems, pharmaceutical software, and other data-driven environments.

Key Features of MySQL

1. **Free & Open Source** – MySQL is free to use under the GPL license.
2. **Cross-Platform** – Works on Windows, Linux, macOS.
3. **Supports Relational Databases** – Uses tables to store data with relationships.
4. **Uses SQL** – Structured Query Language is used to interact with the database.
5. **Security** – Supports user privileges and password encryption.
6. **Fast & Scalable** – Can handle small to very large databases efficiently.
7. **Integration** – Works well with programming languages like PHP, Python, Java, etc.

SQL – Structured Query Language

- SQL is a standard language for accessing and managing databases. It is used in MySQL to perform operations like:

SQL Command	Purpose
CREATE	To create tables or databases
INSERT	To add data into tables
SELECT	To retrieve data
UPDATE	To modify existing data
DELETE	To remove data
DROP	To delete tables or databases
WHERE	To filter data
ORDER BY	To sort data

MySQL in Pharmacy Use

Application	How MySQL Helps
Inventory Management	Track quantity, expiry, and purchase of drugs
Patient Records	Store and retrieve health data securely
Billing Systems	Automate and organize payment details
Drug Database	Quick access to drug names, uses, and prices
Sales Reports	Generate daily/monthly data reports

Advantages of MySQL

- ✓ Easy to learn and use
- ✓ Reliable and secure
- ✓ High performance
- ✓ Widely supported and documented
- ✓ Integration with web and desktop applications
- ✓ Ideal for both small and large scale systems

Introduction to MS Access (Microsoft Access)

- MS Access is a desktop database application that allows users to create, store, retrieve, and manage data using tables, queries, forms, and reports.
- Microsoft Access (MS Access) is a relational database management system (RDBMS) developed by Microsoft. It is part of the Microsoft Office Suite and provides an easy-to-use graphical interface to create, manage, and analyze databases without needing advanced programming skills.

Main Components of MS Access

Component	Description
Tables	Store data in rows and columns (like Excel)
Queries	Retrieve and filter data from one or more tables
Forms	User-friendly interfaces for data entry
Reports	Used for printing or viewing summarized data
Macros	Automate tasks within the database
Modules	For advanced coding using VBA (Visual Basic for Applications)

Features of MS Access

1. **Relational Data Storage** – Stores data in related tables.
2. **User-Friendly Interface** – Uses wizards and drag-and-drop tools.
3. **Forms and Reports** – For easy data entry and visual presentation.
4. **Built-in Query Language (SQL)** – To retrieve and manipulate data.
5. **Integration with MS Office** – Works with Excel, Word, Outlook.
6. **Data Validation** – Ensures accuracy and consistency.
7. **Security** – Provides password protection and user-level access.

Why Use MS Access in Pharmacy/Healthcare

Use Case	Purpose
Drug Inventory System	Track medicine stock, expiry, quantity
Patient Records	Store patient visits, history, medications
Sales and Billing	Generate invoices and receipts
Prescription Management	Manage doctors' prescriptions and doses
Report Generation	Create monthly or yearly data reports

Advantages of MS Access

Benefit	Explanation
Easy to use	No need for advanced programming skills
Built-in Templates	Speeds up database creation
Automation	Using Macros and VBA
Integration	With Excel, Outlook, and Word
Powerful Queries	Easy search and filter of large data
Form and Report Builders	Simplify data entry and presentation

Limitations

- Not suitable for large-scale web-based applications
- Limited concurrent users (ideal for small teams)
- Less scalable compared to MySQL or Oracle

Introduction to Pharmacy Drug Database

- A pharmacy drug database is a computer-based information system that contains detailed data about medications, including drug names, dosage forms, strengths, indications, side effects, interactions, stock, and expiry dates.
- A Pharmacy Drug Database is a structured digital system used to store, manage, retrieve, and update information about pharmaceutical drugs. It is commonly used in hospitals, retail pharmacies, clinics, and research institutions to ensure safe, accurate, and efficient drug management.

Main Components of a Drug Database

Component	Description
Drug ID	Unique identification number/code
Drug Name	Generic or brand name
Dosage Form	Tablet, syrup, injection, etc.
Strength	Amount of active ingredient (e.g., 500 mg)
Indications	Medical conditions the drug is used for
Contraindications	Conditions when the drug should not be used
Side Effects	Known adverse effects
Drug Interactions	Interactions with other drugs
Manufacturer	Company producing the drug
Batch Number	Identification for production lot
Expiry Date	Shelf life of the drug
Stock Quantity	Number of units in pharmacy
Price	Cost per unit or pack

Functions of a Pharmacy Drug Database

1. **Inventory Management** – Track drug stock and generate alerts for low stock or expiry.
2. **Dispensing Records** – Record issued medications to patients.
3. **Drug Information System** – Provide data about uses, side effects, and interactions.
4. **Regulatory Compliance** – Maintain records for audits and legal purposes.
5. **Billing and Invoicing** – Integrated pricing for accurate billing.
6. **Alerts and Warnings** – For contraindications, drug interactions, and expired drugs.

Database Software Commonly Used

Software	Features
MS Access	Easy interface for small clinics or educational use
MySQL/PostgreSQL	Powerful databases for hospitals or pharmacy chains
Oracle DB	Used in advanced clinical/hospital setups
Custom ERP Systems	Built-in drug database modules in pharmacy software

Benefits of a Drug Database in Pharmacy

Advantage	Description
Accurate Drug Tracking	Real-time inventory and expiry updates
Informed Decision-Making	Access to drug profiles, side effects, and interactions
Data Security	Prevents loss or misuse of data
Stock Optimization	Helps avoid overstocking or stock-outs
Easy Record Keeping	Maintains data for regulatory audits
Automated Billing	Reduces human error and speeds up transactions

Challenges in Pharmacy Drug Databases

- Data entry errors if not validated properly
- Requires regular updates and maintenance
- May need training for staff to use effectively
- Connectivity issues in cloud-based systems