www.BrainKart.com

Anna University

for Affilated Engineering College - 2021 Regulation

Information Technology

1st Semester | 2nd Semester | 3rd Semester | 4th Semester

5th Semester | 6th Semester | 7th Semester | 8th Semester

Click on Subject/Paper under Semester to enter.

1st Semester

- Professional English - I - HS3152
- Matrices and Calculus - MA3151
- Engineering Physics - PH3151
- Engineering Chemistry - CY3151
- Problem Solving and Python Programming - GE3151

2nd Semester

- Professional English - II - HS3252
- Statistics and Numerical Methods - MA3251
- Engineering Graphics - GE3251
- Physics for Information Science - PH3256
- Basic Electrical and Electronics Engineering - BE3251
- Programming in C - CS3251

3rd Semester

- Discrete Mathematics - MA3354
- Digital Principles and Computer Organization - CS3351
- Foundation of Data Science - CS3352
- Data Structures and Algorithms - CD3291
- Object Oriented Programming - CS3391

4th Semester

- Environmental Sciences and Sustainability - GE3451
- Theory of Computation - CS3452
- Artificial Intelligence and Machine Learning - CS3491
- Database Management System - CS3492
- Web Essentials - IT3401
- Introduction to Operating Systems - CS3451

5th Semester

- Computer Networks - CS3591
- Full Stack Web Development - IT3501
- Distributed Computing - CS3551
- Embedded Systems and IoT - CS3691
- Elective 1
- Elective 2

6th Semester

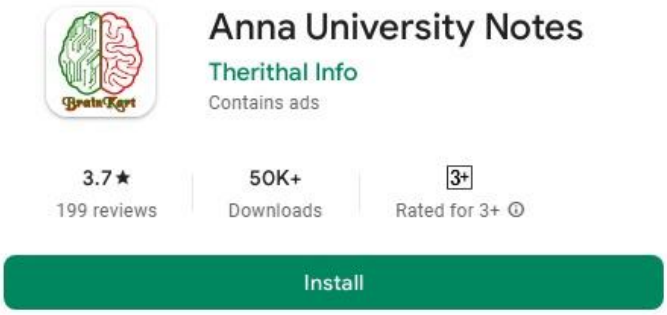
- Object Oriented Software Engineering - CCS356
- Open Elective-1
- Elective-3
- Elective-4
- Elective-5
- Elective-6

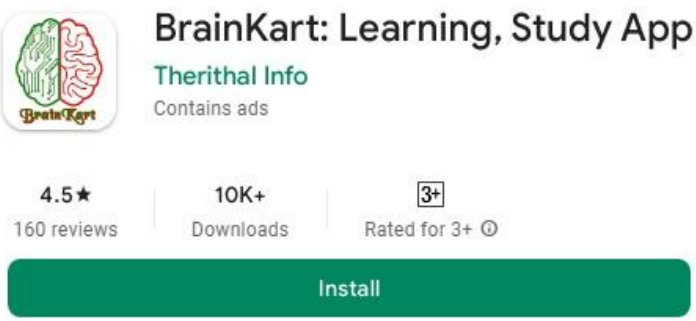
7th Semester

- Human Values and Ethics - GE3791
- Open Elective 2
- Open Elective 3
- Open Elective 4
- Management Elective

8th Semester

- Project Work / Intership





All Computer Engg Subjects - [B.E., M.E.,]		(Click on Subjects to enter)
Programming in C	Computer Networks	Operating Systems
Programming and Data Structures I	Programming and Data Structure II	Problem Solving and Python Programming
Database Management Systems	Computer Architecture	Analog and Digital Communication
Design and Analysis of Algorithms	Microprocessors and Microcontrollers	Object Oriented Analysis and Design
Software Engineering	Discrete Mathematics	Internet Programming
Theory of Computation	Computer Graphics	Distributed Systems
Mobile Computing	Compiler Design	Digital Signal Processing
Artificial Intelligence	Software Testing	Grid and Cloud Computing
Data Ware Housing and Data Mining	Cryptography and Network Security	Resource Management Techniques
Service Oriented Architecture	Embedded and Real Time Systems	Multi - Core Architectures and Programming
Probability and Queueing Theory	Physics for Information Science	Transforms and Partial Differential Equations
Technical English	Engineering Physics	Engineering Chemistry
Engineering Graphics	Total Quality Management	Professional Ethics in Engineering
Basic Electrical and Electronics and Measurement Engineering	Problem Solving and Python Programming	Environmental Science and Engineering

IT3501- Full Stack Web Development

Unit - I

Understanding the Basic Web Development Framework:

Web:

The full form of WWW is the World Wide Web. WWW is also called a Web and it is a catalogue of an order of all websites connected to the worldwide Internet. It is an information system in which linked hypertext data and resources are accessed over the Internet.

In simply it refers to Websites and Web Pages or anything that works over the internet.

Development: Refers to building the application from scratch.

Web Development:

Web development refers to the creating, building, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

Web Development can be classified into two ways:

- Frontend Development

The part of a website where the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application.

Popular Frontend Technologies

- **HTML:** HTML stands for HyperText Markup Language. It is used to design the front end portion of web pages using markup language. It acts as a skeleton for a website since it is used to make the structure of a website.
- **CSS:** Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. It is used to style our website.
- **JavaScript:** JavaScript is a scripting language used to provide a dynamic behavior to our website.
- **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular CSS

framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).

- Bootstrap 4
- Bootstrap 5

Backend Development

Backend is the server side of a website. It is part of the website that users cannot see and interact with. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.

Popular Backend Technologies

- **PHP:** PHP is a server-side scripting language designed specifically for web development.
- **Java:** Java is one of the most popular and widely used programming languages. It is highly scalable.
- **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside a browser.

Full-stack Development

Typically, back-end website development and front-end development are carried out by different professionals with expertise in each.

When the web solution is developed by a single developer who has experience with both front-end and back-end, it is called full-stack development.

Web Development Framework:

A web development framework is a set of resources and tools for software developers to build and manage web applications, web services and websites, as well as to develop application programming interfaces (APIs). Web development frameworks are also referred to as web application frameworks or simply web frameworks.

Web development frameworks enable developers to build applications that can run on well-known technology stacks such as

the Linux, Apache, MySQL and PHP (LAMP) stack. Most frameworks provide a wide range of features and functionality that help streamline application development.

Because web development frameworks are so comprehensive in scope, they offer development teams several important benefits, including the following:

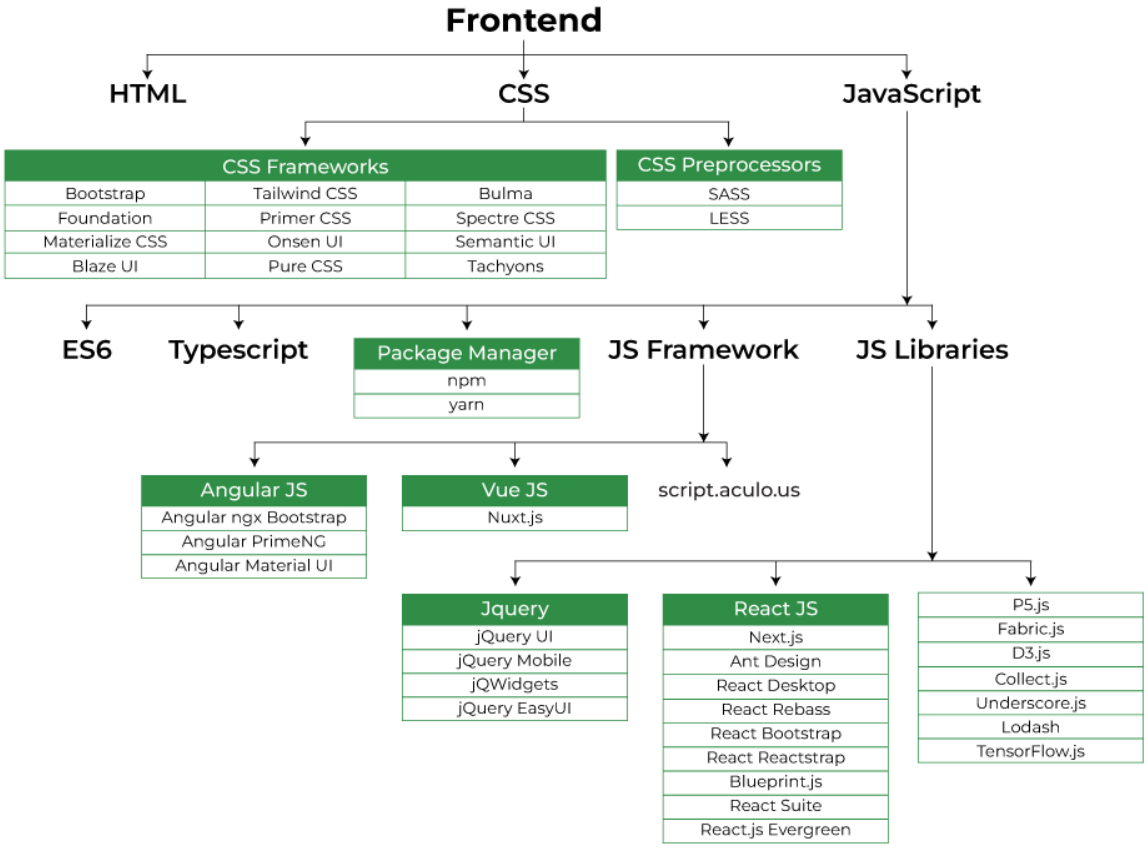
- Developers can build applications faster and more efficiently because they write less code, reuse code, and contend with fewer errors and bugs.
- Many frameworks are open source and backed by strong developer communities that help to optimize the code and reduce errors, leading to better performance and reliability, as well as easier maintenance.
- Many web development frameworks are widely used, adhere to industry standards and are backed by strong developer communities. The continuous vetting and improvements that this provides results in better security. Developers also avoid many of the risks that come with building applications from scratch.
- Web development frameworks speed up application development, reduce errors, simplify debugging and increase reliability. Many of them are also open source and free. Taken together, the factors can significantly reduce overall development costs.

A web development framework also provides the foundation and system-level services necessary to support a content management system (CMS). A content management system is an application built on top of the development framework that adds functionality for dynamically managing digital web content.

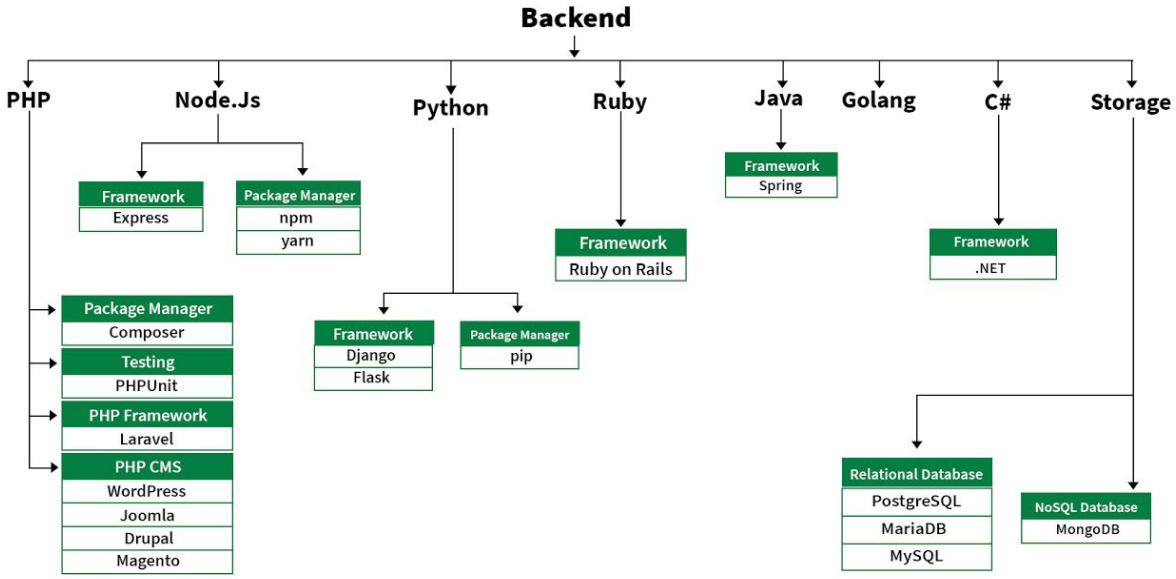
What are the web framework types?

Organizations can choose from a wide range of web development frameworks -- each offering an assortment of features -- giving development teams plenty of options from which to choose. Despite their differences, however, most frameworks fall into one of two categories: those that target front-end development and those that target the back end:

- **Front-end frameworks.** Also called user-side or client-side frameworks, front-end frameworks focus on the user-facing elements of a web application. They provide the components and templates necessary to render passive or interactive webpages in a browser, using industry technologies such as HTML, CSS, JavaScript and jQuery.



- **Back-end frameworks.** These frameworks, also called server-side frameworks, target the server and back-end components that support a web application. They're responsible for mapping URLs, processing HTTP requests, interfacing with data sources and supporting other back-end operations. Back-end frameworks use industry technologies such as Python, PHP, .NET, Java and Ruby.



Web development frameworks are also distinguished from each other by their approach to application architecture. Many web development frameworks are based on a Model-View-Controller (MVC) architecture, which separates the web application into three layers.

- The *Model* layer is concerned with the back-end business logic and data.
- The *View* layer focuses on the user interface and facilitating interactivity.
- The *Controller* layer acts as an interface between the model and view layers, processing the requests between them.

Components of Web Development:

- Application templates for presenting information within a browser.
- Programming environment for scripting the flow of information.
- APIs for accessing back-end data resources
- Code libraries with prebuilt components and code snippets.
- Support for debugging, quality assurance (QA) testing and code reusability.

Benefits of Web Development Framework:

Makes the Development Process Easier

Web frameworks provide pre-written code libraries, modules, and guidelines to developers, which can greatly accelerate the creation process. Assuring scalability, maintainability, and adherence to industry norms for the code also helps.

Eases Debugging and Application Maintenance

Web frameworks give programmers access to tools that simplify managing and debugging their web applications. Frameworks frequently come with built-in debugging tools and can identify and correct typing errors and bugs.

Reduces Code Length

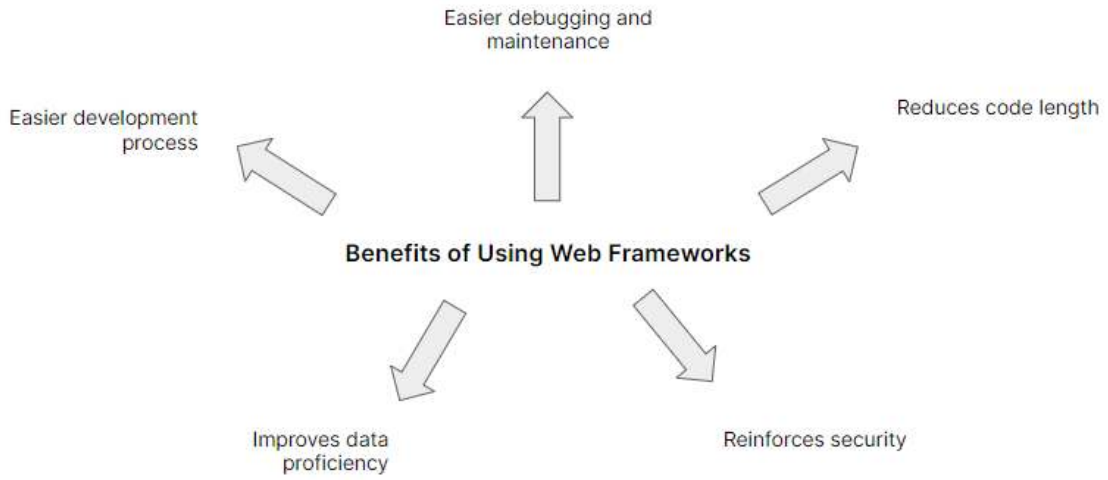
With the aid of web frameworks, developers can achieve more with less code. Pre-built libraries and modules can speed up development by saving developers time and lowering the amount of code they need to create.

Improves Database Proficiency

Web frameworks often include built-in tools for database integration, making it easier to work with databases. This can help improve database proficiency, which is essential for web applications that rely on data.

Reinforces Security

Web frameworks provide developers with built-in security features and guidelines to help reinforce security best practices. This helps ensure that web applications are secure and less vulnerable to attacks.



Users

USER:

Web Application

A web application is software that runs in your web browser. Businesses have to exchange information and deliver services remotely. They use web applications to connect with customers conveniently and securely. The most common website features like shopping carts, product search and filtering, instant messaging, and social media newsfeeds are web applications in their design. They allow you to access complex functionality without installing or configuring software.

Benefits of web applications

Web applications have several benefits, with almost all major enterprises utilizing them as part of their user offerings. Here are some of the most common benefits associated with web apps.

1. **Accessibility**

Web apps can be accessed from all web browsers and across various personal and business devices. Teams in different locations can access shared documents, content management systems, and other business services through subscription-based web applications.

2. **Efficient development**

As detailed, the development process for web apps is relatively simple and cost-effective for businesses. Small teams can achieve short development cycles, making web applications an efficient and affordable method of building computer programs. In addition, because the same version works across all modern browsers and devices, you won't have to create several different iterations for multiple platforms.

3. **User simplicity**

Web apps don't require users to download them, making them easy to access while eliminating the need for end-user maintenance and hard drive capacity. Web applications automatically receive software and security updates, meaning they are always up to date and less at risk of security breaches.

4. **Scalability**

Businesses using web apps can add users as and when they need, without additional infrastructure or costly hardware. In addition, the vast majority of web application data is stored in the cloud, meaning your business won't have to invest in additional storage capacity to run web apps.

Some common web applications

There are numerous types of web applications. Here are some of the most well-known.

- **Workplace collaboration web applications**

Workplace collaboration web apps allow team members to access documents, shared calendars, business instant messaging services, and other enterprise tools.

- **Ecommerce web applications**

Ecommerce web apps such as Amazon.com enable users to browse, search, and pay for products online.

- **Email web applications**

Webmail apps are widely used by enterprises and personal users to access their emails. They often include other communication tools such as instant messaging and video meetings.

- **Online banking web applications**

Business and personal users widely use online banking web apps to access their accounts and other financial products such as loans and mortgages.

- **Technical documentation**

You can use web applications to create and share technical documentation like user manuals, how-to guides and device specifications

Web Server

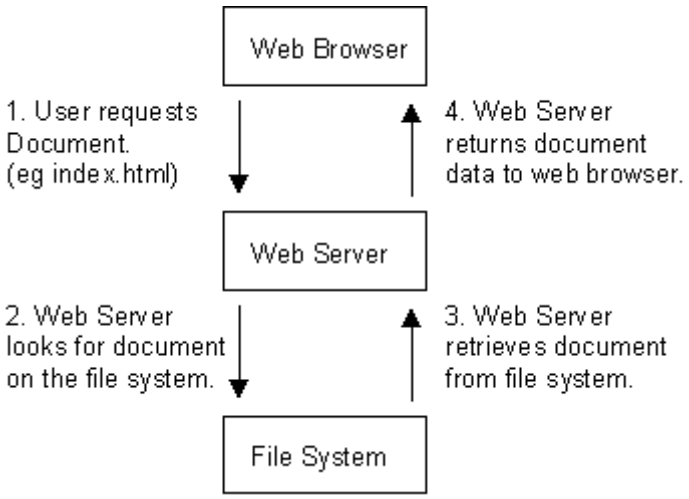
The term *web server* can refer to hardware or software, or both of them working together.

1. On the hardware side, a web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS stylesheets, and JavaScript files). A web server connects to the

Internet and supports physical data interchange with other devices connected to the web.

2. On the software side, a web server includes several parts that control how web users access hosted files. At a minimum, this is an *HTTP server*. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.

At the most basic level, whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct (hardware) web server, the (software) *HTTP server* accepts the request, finds the requested document, and sends it back to the browser, also through HTTP. (If the server doesn't find the requested document, it returns a 404 response instead.)



To publish a website, you need either a static or a dynamic web server.

A **static web server**, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files as-is to your browser.

A **dynamic web server** consists of a static web server plus extra software, most commonly an *application server* and a *database*. We call it "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server.

Web Browser

A software application used to access information on the World Wide Web is called a Web Browser. When a user requests some information, the web browser fetches the data from a web server and then displays the webpage on the user's screen.

Types of Web Browser

The functions of all web browsers are the same. Thus, more than the different types there are different web browsers which have been used over the years.

Discussed below are different web browser examples and their specific features:

1. WorldWideWeb

- The first web browser ever
- Launched in 1990
- It was later named “Nexus” to avoid any confusion with the World Wide Web
- Had the very basic features and less interactive in terms of graphical interface
- Did not have the feature of bookmark

2. Mosaic

- It was launched in 1993
- The second web browser which was launched
- Had a better graphical interface. Images, text and graphics could all be integrated
- It was developed at the National Center for Supercomputing Applications
- The team which was responsible for creating Mosaic was lead by Marc Andreessen
- It was named “the world’s first popular browser”

3. Netscape Navigator

- It was released in 1994
- In the 1990s, it was the dominant browser in terms of usage share
- More versions of this browser were launched by Netscape

- It had an advanced licensing scheme and allowed free usage for non-commercial purposes

4. Internet Explorer

- It was launched in 1995 by Microsoft
- By 2003, it has attained almost 95% of usage share and had become the most popular browsers of all
- Close to 10 versions of Internet Explorer were released by Microsoft and were updated gradually
- It was included in the Microsoft Windows operating system
- In 2015, it was replaced with "Microsoft Edge", as it became the default browser on Windows 10

5. Firefox

- It was introduced in 2002 and was developed by Mozilla Foundation
- Firefox overtook the usage share from Internet Explorer and became the dominant browser during 2003-04
- Location-aware browsing was made available with Firefox
- This browser was also made available for mobile phones, tablets, etc.

6. Google Chrome

- It was launched in 2008 by Google
- It is a cross-platform web browser
- Multiple features from old browsers were amalgamated to form better and newer features
- To save computers from malware, Google developed the ad-blocking feature to keep the user data safe and secure
- Incognito mode is provided where private searching is available where no cookies or history is saved
- Till date, it has the best user interface

Apart from these, Opera Mini web browser was introduced in 2005 which was specially designed for mobile users. Before the mobile version, the computer version "Opera" was also released in 1995. It supported a decent user interface and was developed by Opera Software.

S. No.	Parameters	Web Browser	Web Server
1.	**Basics**	Web Browser is an Application program that displays a World wide web document. It usually uses the internet service to access the document.	A web server is a program or the computer that provide services to other programs called client.
2.	**Function**	The Web browser requests the server for the web documents and services.	The Web server accepts, approves, and responds to the request made by the web browser for a web document or service.
3.	**Responsibility**	A web browser is a programme that uses websites to search the internet for information.	The web server is responsible for connecting websites and web browsers.
4.	**Interface**	The web browser acts as an interface between the server and the client and displays a web document to the client.	The web server is a software or a system which maintain the web applications, generate response and accept clients data.
5.	**Components of architecture**	Components of web browser architecture- a controller, client program, and interpreters.	Components of web server architecture- hardware, operating system software, and Web server software.
6.	**HTTP request and response**	The web browser sends an HTTP request and gets an HTTP response.	The web server gets HTTP requests and sends HTTP responses.
7.	**Processing Model**	Doesn't exist any processing model for the web browser.	There exist three types of processing models for web servers i.e Process-based, Thread based, and Hybrid.
8.	**Storing data**	Web browser stores the	Web servers provide an area to

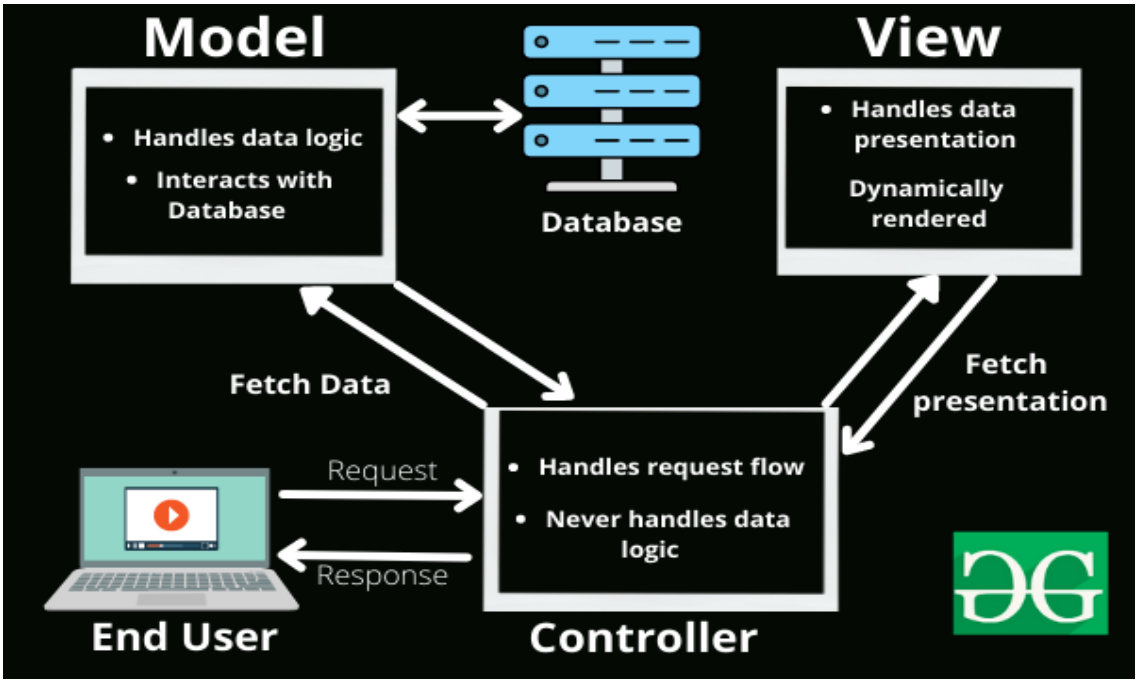
S. No.	Parameters	Web Browser	Web Server
		cookies for different websites.	store and organize the pages of the website.
9.	**Installation**	The web browser is installed on the client's computer.	The web server can be a remote machine placed on the other side of your network or even on the other end of the globe, or it is your very own personal computer at home.
10.	**Examples**	Examples of Web browsers are Mozilla Firefox, Google Chrome, and Internet Explorer.	An example of a Web Server is Apache Server.

MVC Architecture:

MVC is an architectural pattern that is used to divide the application into three components, namely - Model, View, and Controller. Separating the application into these three components makes it easier to scale the application and makes it more extensive and easier to maintain.

Features of MVC :

- It provides a clear separation of **business logic, Ul logic, and input logic.**
- It offers full control over your HTML and URLs which makes it easy to design web application architecture.
- It is a powerful URL-mapping component using which we can build applications that have comprehensible and searchable URLs.
- It supports **Test Driven Development (TDD).**



Components of MVC :

The MVC framework includes the following 3 components:

Controller:

- The controller is the component that enables the interconnection between the views and the model so it acts as an intermediary. The controller doesn't have to worry about handling data logic, it just tells the model what to do. It process all the business logic and incoming requests, manipulate data using the **Model** component and interact with the **View** to render the final output.

View:

- The **View** component is used for all the UI logic of the application. It generates a user interface for the user. Views are created by the data which is collected by the model component but these data aren't taken directly but through the controller. It only interacts with the controller.

Model:

- The **Model** component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. It can add or retrieve data from the database. It responds to the controller's request because the controller can't interact with the database by itself. The model

interacts with the database and gives the required data back to the controller.

Advantages of MVC:

- Codes are easy to maintain and they can be extended easily.
- The MVC **model** component can be tested separately.
- The components of MVC can be developed simultaneously.
- It reduces complexity by dividing an application into three units. **Model, view, and controller.**
- It supports **Test Driven Development (TDD).**
- It works well for Web apps that are supported by large teams of web designers and developers.
- This architecture helps to test components independently as all classes and objects are independent of each other
- **Search Engine Optimization (SEO)** Friendly.

Disadvantages of MVC:

- It is difficult to read, change, test, and reuse this model
- It is not suitable for building small applications.
- The inefficiency of data access in view.
- The framework navigation can be complex as it introduces new layers of abstraction which requires users to adapt to the decomposition criteria of MVC.
- Increased complexity and Inefficiency of data

Popular MVC Frameworks:

Some of the most popular and extensively used MVC frameworks are listed below.

- Ruby on Rails
- Django
- CherryPy
- Spring MVC
- Catalyst
- Rails
- Zend Framework
- Fuel PHP
- Laravel
- Symphony

Understanding of Different Stack:

Express

The Express module acts as the webserver in the Node.js-to-Angular stack. The fact that it is running in Node.js makes it easy to configure, implement, and control. The Express module extends Node.js to provide several key components for handling web requests.

This allows you to implement a running webserver in Node.js with only a few lines of code. For example, the Express module provides the ability to easily set up destination routes (URLs) for users to connect to.

It also provides great functionality on working with the HTTP request and response objects, including things like cookies and HTTP headers. The following is a partial list of the valuable features of Express:

■ Route management: Express makes it easy to define routes (URL endpoints) that tie directly to Node.js script functionality on the server.

■ Error handling: Express provides built-in error handling for documents not found and other errors.

■ Easy integration: An Express server can easily be implemented behind an existing reverse proxy system such as Nginx or Varnish. This allows it to be easily integrated into your existing secured system.

■ Cookies: Express provides easy cookie management.

■ Session and cache management: Express also enables session management and cache management.

Angular

Angular is a client-side framework developed by Google. Angular provides all the functionality needed to handle user input in the browser, manipulate data on the client side, and control how elements are displayed in the browser view. It is written using TypeScript.

The entire theory behind Angular is to provide a framework that makes it easy to implement web applications using the MVC framework.

Other JavaScript frameworks could be used with the Node.js platform, such as Backbone, Ember, and Meteor.

However, Angular has the best design, feature set, and trajectory at this writing.

Here are some of the benefits of Angular:

■ Data binding: Angular has a clean method to bind data to HTML elements using its powerful scope mechanism.

■ Extensibility: The Angular architecture allows you to easily extend almost every aspect of the language to provide your own custom implementations.

■ Clean: Angular forces you to write clean, logical code.

■ Reusable code: The combination of extensibility and clean code makes it easy to write reusable code in Angular. In fact, the language often forces you to do so when creating custom services.

■ Support: Google is investing a lot into this project, which gives it an advantage over other similar initiatives.

■ Compatibility: Angular is based on TypeScript, which makes it easier to begin integrating Angular into your environment and to reuse pieces of your existing code within the structure of the Angular framework.

Node.js

Node.js is a development framework based on Google's V8 JavaScript engine. Therefore, Node.js code is written in JavaScript and then compiled into machine code by V8 to be executed.

Many of your backend services can be written in Node.js, as can the server-side scripts and any supporting web application functionality. The nice thing about Node.js is that it is all just JavaScript, so you can easily take functionality from a client-side script and place it in a server-side script.

Also, the web server can run directly within the Node.js platform as a Node.js module, so it makes it much easier than, say, Apache at wiring up new services or server-side scripts.

The following are just a few reasons why Node.js is a great framework to start from:

■ JavaScript end-to-end: One of the biggest advantages to Node.js is that it allows you to write both server- and client-side scripts in JavaScript. There have always been difficulties in deciding where to put scripting logic. Too much on the client side makes the client cumbersome and unwieldy, but too much on the server side slows down web applications and puts a heavy burden on the webserver. With Node.js you can take JavaScript written on the client and easily adapt it for the server and vice versa. Also, your client developers and server developers will be speaking the same language.

■ Event-driven scalability: Node.js applies a different logic to handling web requests. Rather than having multiple threads waiting to process web requests, they are processed on the same thread using a basic event model. This allows Node.js webservers to scale in ways that traditional webservers never can. This is discussed in more detail in later chapters.

■ Extensibility: Node.js has a great following and an active development community. New modules to extend Node.js functionality are being developed all

the time. Also it is simple to install and include new modules in Node.js, making it easy to extend a Node.js project to include new functionality in minutes.

■ Time: Let's face it, time is valuable. Node.js is super easy to set up and develop in. In only a few minutes, you can install Node.js and have a working webserver.

MongoDB

MongoDB is an agile and scalable NoSQL database. The name Mongo comes from "humongous."

It is based on the NoSQL document store model, meaning that data is stored in the database as a form of JSON objects rather than the traditional columns and rows of a relational database.

MongoDB provides great website backend storage for high traffic websites that need to store data such as user comments, blogs, or other items because it is fast, scalable, and easy to implement.

This book covers using the MongoDB driver library to access MongoDB from Node.js. Node.js supports a variety of DB access drivers, so the data store could just as easily be MySQL or some other database. However, the following are some of the reasons that MongoDB really fits in the Node.js stack well:

■ Document orientation: Because MongoDB is document-oriented, the data is stored in the database in a format close to what you will be dealing with in both server-side and clientside scripts. This eliminates the need to transfer data from rows to objects and back. Understanding the Node.js-to-Angular Stack Components 13

■ High performance: MongoDB is one of the highest performing databases available. Especially today when more and more people interact with websites, it is important to have a backend that can support heavy traffic.

■ High availability: MongoDB's replication model makes it easy to maintain scalability while keeping high performance.

■ High scalability: MongoDB's structure makes it easy to scale horizontally by sharing the data across multiple servers.

■ No SQL injection: MongoDB is not susceptible to SQL injection (putting SQL statements in web forms or other input from the browser that compromises the DB security) because objects are stored as objects, not using SQL strings.

React

React anchors the MERN stack. In some sense, it is the defining component of the MERN stack.

React is an open-source JavaScript library maintained by Facebook that can be used for creating views rendered in HTML.

Unlike AngularJS, React is not a framework. It is a library. Thus, it does not, by itself, dictate a framework pattern such as the MVC pattern.

www.BrainKart.com

Anna University

for Affilated Engineering College - 2021 Regulation

Information Technology

1st Semester | 2nd Semester | 3rd Semester | 4th Semester

5th Semester | 6th Semester | 7th Semester | 8th Semester

Click on Subject/Paper under Semester to enter.

1st Semester

- Professional English - I - HS3152
- Matrices and Calculus - MA3151
- Engineering Physics - PH3151
- Engineering Chemistry - CY3151
- Problem Solving and Python Programming - GE3151

2nd Semester

- Professional English - II - HS3252
- Statistics and Numerical Methods - MA3251
- Engineering Graphics - GE3251
- Physics for Information Science - PH3256
- Basic Electrical and Electronics Engineering - BE3251
- Programming in C - CS3251

3rd Semester

- Discrete Mathematics - MA3354
- Digital Principles and Computer Organization - CS3351
- Foundation of Data Science - CS3352
- Data Structures and Algorithms - CD3291
- Object Oriented Programming - CS3391

4th Semester

- Environmental Sciences and Sustainability - GE3451
- Theory of Computation - CS3452
- Artificial Intelligence and Machine Learning - CS3491
- Database Management System - CS3492
- Web Essentials - IT3401
- Introduction to Operating Systems - CS3451

5th Semester

- Computer Networks - CS3591
- Full Stack Web Development - IT3501
- Distributed Computing - CS3551
- Embedded Systems and IoT - CS3691
- Elective 1
- Elective 2

6th Semester

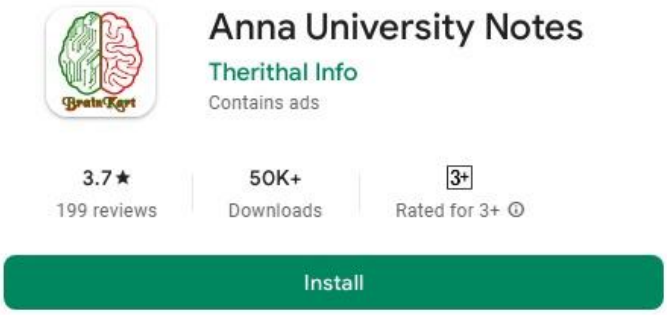
- Object Oriented Software Engineering - CCS356
- Open Elective-1
- Elective-3
- Elective-4
- Elective-5
- Elective-6

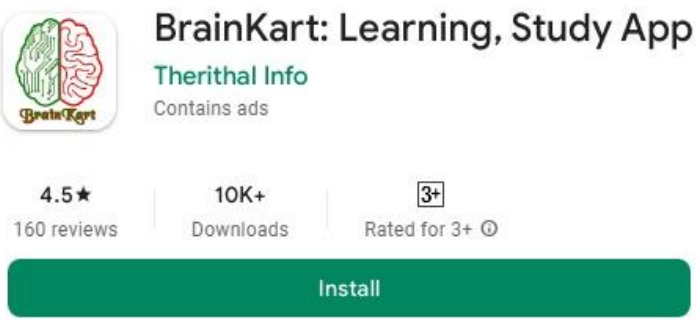
7th Semester

- Human Values and Ethics - GE3791
- Open Elective 2
- Open Elective 3
- Open Elective 4
- Management Elective

8th Semester

- Project Work / Intership





All Computer Engg Subjects - [B.E., M.E.,]		(Click on Subjects to enter)
Programming in C	Computer Networks	Operating Systems
Programming and Data Structures I	Programming and Data Structure II	Problem Solving and Python Programming
Database Management Systems	Computer Architecture	Analog and Digital Communication
Design and Analysis of Algorithms	Microprocessors and Microcontrollers	Object Oriented Analysis and Design
Software Engineering	Discrete Mathematics	Internet Programming
Theory of Computation	Computer Graphics	Distributed Systems
Mobile Computing	Compiler Design	Digital Signal Processing
Artificial Intelligence	Software Testing	Grid and Cloud Computing
Data Ware Housing and Data Mining	Cryptography and Network Security	Resource Management Techniques
Service Oriented Architecture	Embedded and Real Time Systems	Multi - Core Architectures and Programming
Probability and Queueing Theory	Physics for Information Science	Transforms and Partial Differential Equations
Technical English	Engineering Physics	Engineering Chemistry
Engineering Graphics	Total Quality Management	Professional Ethics in Engineering
Basic Electrical and Electronics and Measurement Engineering	Problem Solving and Python Programming	Environmental Science and Engineering