SUBMITTED BY

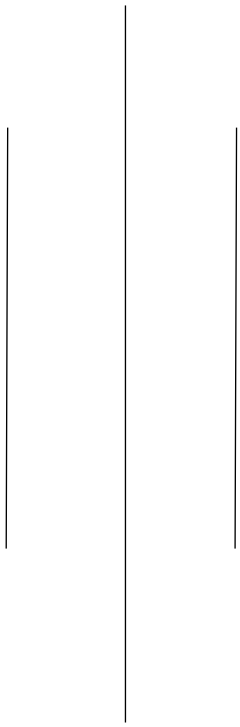
FULL NAME: Gaurav Shakya

COURSE: BTEC HND in Computing

SEMESTER: Second Semester

SECTION: C

YEAR OF ENROLLMENT: 2018



SUBMITTED TO

Mr. Harish Thaguna

Contents

Part 1

Scenario 1:

Web Tech Pvt. Ltd. is a leading software company along with web hosting and domain registering company since 2012. The company is in healthy condition in terms of software development and in providing other services. The company has decided to hire some new employees to the team. You are one of them. Being fresher to the company, they want to know your knowledge in terms of web hosting details and familiarization with website technologies.

1. Prepare a report identifying purpose and types of DNS along with how domain names are organized and managed. You should also explain the purpose and relationships between communication protocols, server hardware, operating systems and web server software with regards to web designing, publishing and accessing a website. Your report should also contain the influence of search engines on website performance along with website crawling, indexing and ranking and provide evidence based support for improving site's index value and rank through search engine optimization.

2. Write an article evaluating the impact of common web development technologies and frameworks with regards to website design, functionality and management. Also, explain the capabilities and relationships between front-end and back-end website technologies along with their relation to presentation and application layers. You should point out the differences between online website creation and custom built sites with regards to design flexibility, performance, functionality, user experience (UX) and user interface (UI).

3. Evaluate a range of tools and techniques available to design and develop a custom built website along with justification to the tools and techniques chosen to realize a custom built website.

Introduction

This is part I, as the company wants to know our knowledge in terms of web hosting details and other website technologies, we are describing about DNS along with how they are organized and managed.

Domain Name System (DNS)

History of Domain Name System (DNS)

ARPANET (Advanced Research Projects Agency Network) is the Internet's grandfather, which was established in 1966. It was built to connect research centers across the United States in order to share information faster. As the years passed by, number of centers linked through it multiplied. As more computer accessed ARPANET, it became harder for the workers to maintain, while numerical IP addresses were getting more complex for the users to remember.

Then a fresh Computer Science graduate named Paul Mockapetris brought his concept of developing the idea of efficient file systems for computers. With the help of Jon Postel and Zaw-Sing Su, he proposed a new nomenclature for websites in 1983. He suggested that host name should include:

1. Name – for example, "IBM"
2. Categories/Purpose – for example, ".com – for commercial purposes."

After a year, the categories were created including '.edu', '.net', '.gov', '.org', etc.

What is Domain Name System (DNS)?

DNS server is a Server that provides Internet Protocol (IP) address to the websites. For example: just like the phone book on your mobile, you need to find Gaurav, so you search "Gaurav", and don't need to remember his actual number, similarly, a user goes to a certain website www.facebook.com where IP is given as 192.168.19.20. So, DNS server are

used by web hosting companies. Similarly, humans access data online through domain names like www.google.com. Web browsers move through Internet Protocol (IP) addresses. Domain names are formed by the rules and procedures of Domain Name System (DNS). Any name registered in the DNS is a domain name. Domain names basically used in various networking contexts and for application-specific naming and addressing purpose.

Source: en.wikibooks.org/wiki/Communication_Networks/DNS

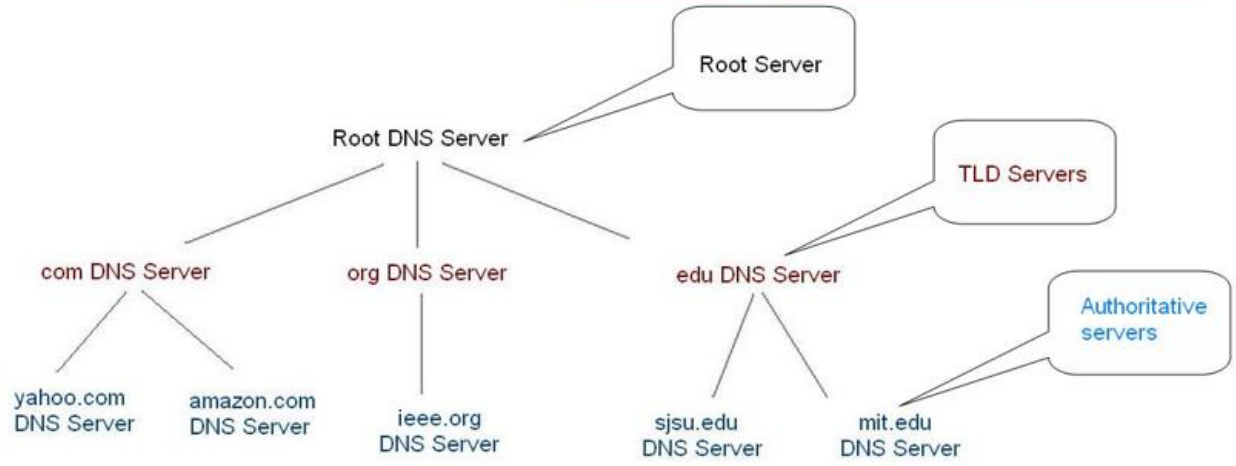


Fig: Structure of Domain Name System (DNS)

What is Domain Management and how it works?

Domain management or domain name management refers to the ongoing tasks of keeping a personal or corporate domain (or domains) stable, secure, and able to support related websites. The domain name not only establishes its owner's presence on the Internet, it also serves as a portal to a business or personal website designed to serve that owner's unique needs. Site owners may be tempted to focus all their attention on setting up and maintaining the website itself, but managing the domain is a key factor in keeping the website live and accessible.

Individual users and small businesses might find that a single domain name is all that's needed to establish an online presence and develop a brand. Larger corporations and those with a number of different business interests may need multiple domains. Many registrars recommend buying as many related domain names as possible to keep them from being used by competitors and to capture all possible variants that might be typed in during a search—including misspellings. That can result in a large portfolio of domain names, some of which

may never be used. Whether a user buys one domain or many, though, ongoing management keeps them updated and working to support the websites they serve.

Who manages DNS?

Due to the importance of the Domain Name System (DNS), management and maintenance is required. While the Internet has no central government, the assignment of "space" (IP, DNS, and ASN) on the Internet is strictly managed. The Domain Name System is managed by the following organizations:

1. (ICANN) Internet Corporation for Assigned Names and Numbers
2. (IANA) Internet Assigned Numbers Authority
3. Operators of Top Level Domains (like Verisign).
4. Accredited registrars (like GoDaddy).

Source: https://www.internet-guide.co.uk/domain-names.html

Fig: ICANN's management of the DNS, and Verisign's of the .com registry has remained controversial

How you publicly available the websites?

Here are the steps for registering a domain:

1. **Choose your registrar or web hosting service:** Remember to check their hosting policies and pricing. ICANN has a list of reputable domain registers.
2. **Search for a domain name:** Your chosen registrar should have an availability checker, such as the one from Bluehost, which you can use to determine if a domain name is available. The advantage of the availability checker is that it will give you suggestions for alternative domains if yours is already taken.
3. **Fill out the forms:** Expect that you will be asked for your personal information and contact details to complete your profile for the “WHOIS” database. Choose whether you want your profile to be public or private.

4. **Pay for your domain:** Prices for the domain names will come up during the availability check. Depending on which web host or registrar you choose, you may be able to get your domain for free by purchasing a hosting package. After payment, you can use your domain name right away or park it until you're ready to launch your own website.
5. **Link domain and website:** Using the domain manager, enter the name of your web host's primary and secondary servers into your registrar's settings for your domain. This process will link your domain name to your website.
6. **Park your domain if you don't have a website:** If you don't have a website yet, your domain registrar can park your domain name temporarily using a website setup for you. Parking a domain means having reserved that particular website address until you have your own website created to use with the name.

It is important to point out that some host and registrars will only register domain names with certain TLDs, so you may need to check their options before signing up.

Communication Protocol

Communications devices have to agree on various physical aspects of data to be exchanged before successful transmission takes place. So, communication protocol are rules defining the communication transmissions.

There may be many properties of it. Most common include: packet size, transmission speed, error correction types, address mapping, flow control, packet sequence, routing, etc.

Web Communication Protocols

People browse the internet by using a web browser. A web browser is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource is identified by Uniform Resource Identifier (URL) and may be a web page, image, video etc.

A web browser lets you to communicate with web servers around the World and giving you all the information with just some clicks. Different web browsers may have different way of retrieving information, but the common one is web communication protocols.

Some popular web communication protocols are: File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), etc.

HyperText Transfer Protocol (HTTP)

HyperText Transfer Protocol (HTTP) is the most widely used web communications protocol so far. If you look in the Address field of your web browser right now, it's likely you'll see "http://" at the front.

HTTP is a classic "client-server" protocol. Users click a link on their web browser (the client), and the browser sends a request over the internet to a web server that houses the site the user requested. The server sends back the content of the site, such as text and images, which display in users' web browsers. HTTP is an unsecure communications protocol because the data it sends back and forth between a browser and a server is unencrypted and can be intercepted by third parties.

Hypertext Transfer Protocol Secure (HTTPS)

HTTPS is similar to HTTP, but different in that it combines with a security protocol called SSL/TLS to provide secure client-server communications over unsecure networks such as the internet. You're most likely to see HTTPS protocols on ecommerce websites that ask for personal financial information like credit card numbers. You know a website is using HTTPS protocols when you see the "https://" in the web address displayed in your browser's Address field.

File Transfer Protocol (FTP)

File Transfer Protocol (FTP) is primarily used to transfer files such as documents, images, music, etc., between remote computers. Users have to log on to an FTP server either through a command line interface or through one of the many FTP graphical client programs available. Once logged on, users can navigate through the remote server's file structure, moving, renaming, deleting, and copying files as if it were their own computer.

Telnet

Telnet is one of the oldest communication protocols. Like HTTP, a Telnet client is used to access remote servers. However, unlike HTTP, where you only request specific files, Telnet is used to actually log on to the remote server and perform functions as if you were

sitting in front of the server terminal. Telnet is rarely used now since it is an unsecure protocol that does not encrypt data sent between remote computers.

IP Security (IPSec)

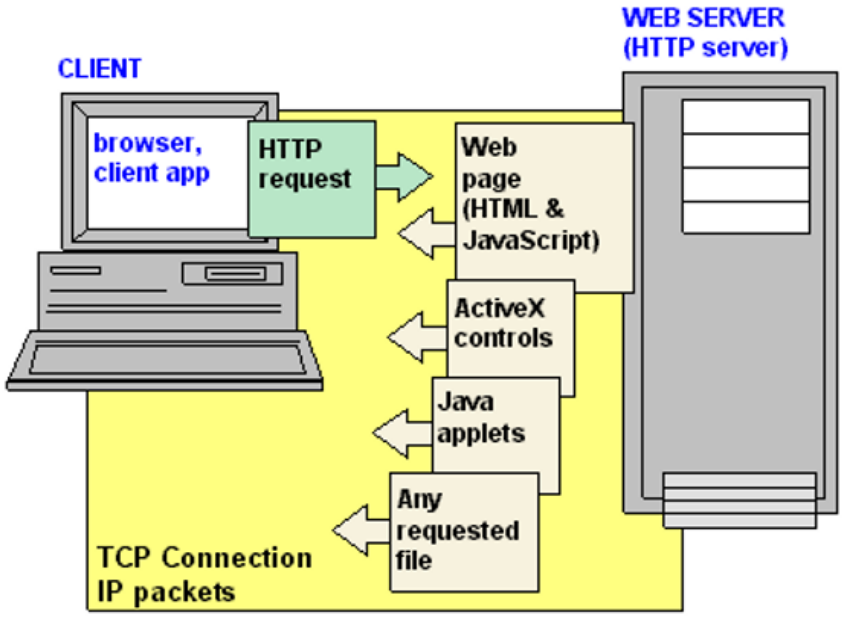
IP Security (IPSec) protocols encrypt packets of data and send them between two computers that share the same cryptographic keys. In other words, the IPSec protocol is like a hallway with no doors linking two rooms. The only place the data can go is between those two rooms. IPSec protocols are used in Virtual Private Networks, which allow employees of a company to log on to their company's secure network through a public network (e.g. from home or a coffee shop.)

Server Hardware

Servers are computers especially designed to respond any request from the users or other computers. The main purpose of Server is to be there for you, to share data and keep in touch with you. Apparently, serer host lies here. Some examples of server tasks are:

1. Email
2. File Sharing
3. Web hosting
4. Network management
5. Custom applications

Source: www.pcmag.com/encyclopedia/term/54342/web-server



Web server providing data and resource to the client PC.

Web Server Software

Web Server Software is basically a platform to run local Web Server. We used xampp because it is a free and open source package consisting of Apache HTTP server, mySql database and interpreters for scripts written in PHP programming languages.

Evaluating the impact of common web development technologies and frameworks with regards to website design, functionality and management.

Web development:

The web development can be defined as the coding or programming that enables website functionality and develop websites for hosting via internet, as per the owner's needs. The web development hierarchy is mentioned below:

1. Client-side coding
2. Server-side coding
3. Database side/technology

Client-Side coding: Client-side means simply all the actions taking place on the user's computer i.e. client PC. Here, the front end and script are visible among the users and it doesn't need interaction with servers. HTML, CSS, JavaScript, etc are the languages of it. Example: In facebook.com, the HTML, CSS and JavaScript that decides how the Facebook page appears to the user are interpreted by the browser on the client side.

Server-Side coding: Server-side are those action that takes place on a web server. Here, it works in the back end which means it couldn't be visible at the client end. Example: In Facebook, we create new account which includes our information. We cannot see where it is saved and how it is saved. All these information are stored in database servers. PHP, ASP.net, Ruby on Rails, Python, etc are the languages involved in it.

Database Side: The database is a simple repository for your data. It stored all user logins and other data that the application will need to retrieve and use throughout its lifetime. Databases can be stored on the same server as the server/back-end component of your application, however, this practice is not secure and should not be used beyond a simple proof-of-concept for your idea. For database coding, Oracle, MS SQL Server, MySQL, etc can be used.

Web Development Technology

Web development technology are the languages that are used to develop websites

The following are the Popular Web Development Technologies that are being widely used for Front-End and Back-End Development:

Front End Web Development

HTML, CSS are the major building blocks for web development. Their Frameworks like Bootstrap and Materialize are most widely used in web development. For Twitter, Bootstrap is popular for its Responsive design and Flexbox and CSS Grid System are also trending topics in CSS, and they are used without any framework to develop responsive websites.

Another trending development technology is Motion UI. It keeps simplicity to a site. Web developers are using its animations that will allow adding styling and make your site unique among thousands of others with static UI.

JavaScript and its Front-End Frameworks like Angular and React and Vue.js are popular frameworks that need to be used in your project.

Back End Web Development

Server-Side Languages/Technologies: Node JS, Python and PHP are the languages for Web Development in 2018. The other languages used in web development are Ruby, C# & ASP.NET

Database: MongoDB is currently popular, with its non-relational database, MySQL is popular relational database. Oracle, SQL Server, PostgreSQL, Firebase are other database systems used in web development.

Back-End Frameworks: Express is the popular frame work for JavaScript. Adonis and Hapi.js are also best frameworks in JavaScript. Frameworks for PHP like Laravel, CodeIgniter, Symphony and Yii2 are popular. For Python- Django, Flask, Web2py are widely used frameworks, and for Ruby – Ruby on rails, and for C# – .NET are best frameworks used.

Web Development Platform/Framework:

A web development framework is a tools for software development to build and manage web applications, web services, and websites. Here are the some of the framework you can work on:

1. Sublime Text

Sublime Text is one that includes well-designed, super-efficient, and ultra-speedy features user interface. There are several that do this well, but arguably the best (and most popular) is Capably run by a one-man development team, the secret program for lofty success lies in the vast array of keyboard shortcuts - such as the ability to simultaneously edit (make the same interactive changes across multiple selected areas) as well as accelerate Navigation. Files, Icons and Lines.

2. Chrome Developer Tools

It would be great if you could edit your HTML and CSS in real time, or debug your JavaScript, looking at the overall performance analysis of your website.

Google's built-in Chrome developer tool lets you do just that. Bundled and available in both Chrome and Safari, they allow developers to enter the internals of their web applications. On top of that, a palette of network tools can help optimize your loading flow, while a timeline gives you a deeper understanding of what the browser is doing at any given time.

3. JQuery

JavaScript has long been considered an essential front-end language by developers, although it’s not without its problems: riddled with browser inconsistencies, its somewhat complicated and unapproachable syntax meant that functionality often suffered.

That was until 2006, when jQuery – a fast, small, cross-platform JavaScript library aimed at simplifying the front-end process – appeared on the scene. By abstracting a lot of the functionality usually left for developers to solve on their own, jQuery allowed greater scope for creating animations, adding plug-ins, or even just navigating documents.

4. GitHub

In GitHub, by rolling out your project with the service, you can view any changes you’ve made or even go back to your previous. The repository hosting service also boasts a rich open-source development community (making collaboration between teams as easy as pie),

as well as providing several other components such as bug tracking, feature requests, task management, and wikis for every project.

Search Engine

Search engine can be defined as a service or program that allows Internet users to search for content via WWW. Example: Whenever you search something in search engines like Google, Bing, etc, they show you contents like text, images, videos etc. This provides a best list of results based on your keywords you trying to search.

How Search Engine works?

Because large search engines contain millions and more pages, many search engines not only search the page but also display the results depending on their needs. This importance is commonly determined by using algorithms. Search engines work by crawling hundreds of billions of pages using their own web crawlers. These web crawlers are commonly referred to as **search engine bots** or **spiders**. A search engine navigates the web by downloading web pages and following links on these pages to discover new pages that have been made available.

What is web crawling?

A crawler is a program that visits Web sites and reads their pages and other information in order to create entries for a search engine index. The major search engines on the Web all have such a program, which is also known as a "spider" or a "bot." Crawlers are typically programmed to visit sites that have been submitted by their owners as new or updated. Entire sites or specific pages can be selectively visited and indexed. Crawlers apparently gained the name because they crawl through a site a page at a time, following the links to other pages on the site until all pages have been read.

These boats are almost always powered by search engines. By applying a search algorithm to the data collected by web crawlers, search engines can provide relevant links in response to user search queries, generating a list of webpages that allow the user to search in Google or Bing (or any other search engine) show after search. .

A web crawler bot is like someone who goes through all the books in a cluttered library and puts together a card catalog so that anyone who visits the library can quickly and easily find those information that they need. To help classify and sort library books by subject, the

organizer will find out what it is about by reading each book's title, summary, and some internal text.

However, unlike a library, the Internet is not made up of physical stacks of books, and this makes it difficult to tell if all the necessary information is properly indexed, or if it is being ignored in large quantities. To try to find all the relevant information that the Internet has to offer, a web crawler bot will start with a fixed set of known webpages and then follow hyperlinks from those pages to other pages, in addition to those other pages follow hyperlinks to pages, and so on.

It is unknown how much of the publicly available Internet has actually been crawled by search engine bots. Some sources estimate that only 40–70% of the Internet is indexed for search - and that there are billions of webpages.

What is search indexing?

We can say, search indexing is just like creating a library card classify for the Internet so that a search engine knows how to find information when someone searches on the Internet. Similarly, it can also be compared to the index on the back of a book, which lists all the places in the book where a certain topic or phrase is mentioned.

The index focuses mostly on the text that appears on the page, and on the metadata about the page that users do not see. When most search engines index a page, they add all the words on the page to the index, except for words like "a," "an," and "the" in the case of Google. When users search for those words, the search engine goes through their index of all the pages where those words appear and select the most relevant ones.

In the context of search indexing, metadata is data that tells search engines what a webpage is about. Often Meta titles and Meta descriptions are those that will appear on search engine result pages, as opposed to the content of the webpage that users see.

The index includes all the discovered URLs along with a number of relevant key signals about the contents of each URL such as:

- The keywords discovered within the page’s content – what topics does the page cover?
- The type of content that is being crawled (using microdata called Schema) – what is included on the page?

- The freshness of the page – how recently was it updated?
- The previous user engagement of the page and/or domain – how do people interact with the page?

How do web crawlers work?

We know that the Internet is extremely changing and expanding. It is due to millions of webpages on the, web crawler bots start from a seed, or a list of known URLs. They crawl the webpages at those URLs first. As they crawl those webpages, they will find hyperlinks to other URLs, and they add those to the list of pages to crawl next.

Given the huge number of webpages on the Internet that can be indexed for search, this process can go on for a long period of time. However, a web crawler will follow certain policies that make it more selective about which pages to crawl, in which order to crawl them, and how often to re-crawl them to check for content updates needed.

The relative importance of each webpage: Most web crawlers don't crawl the entire publicly available Internet and aren't intended to; instead they decide which pages to crawl first based on the number of other pages that link to that page, the amount of visitors that page gets, and other factors that signify the page's likelihood of containing important information.

The idea is that a webpage that is cited by a lot of other webpages and gets a lot of visitors is likely to contain high-quality, authoritative information, so it's especially important that a search engine has it indexed – just as a library might make sure to keep plenty of copies of a book that gets checked out by lots of people.

Revisiting webpages: Contents on the Web server always update, remove, or move to new locations. Web crawlers will periodically need to revisit pages to make sure the latest version of the content is indexed.

Robots.txt requirements: Web crawlers can also make decision which pages to crawl based on the robots.txt protocol (also known as the robots exclusion protocol). They will check the robots.txt file hosted by that page's web server before crawling a webpage. A robots.txt file is a text file that specifies the rules for any bots accessing the hosted website or application. These rules define which pages the bots can crawl, and which links they can follow.

All these factors are weighted differently within the proprietary algorithms that each search engine builds into their spider bots. Web crawlers from different search engines will behave slightly differently, although the end goal is the same: to download and index content from webpages.

W3C Validation:

W3C stands for the World Wide Web Consortium, who are the recognized standards organization for the World Wide Web.

The organization was founded by Tim Berners-Lee and is run by a full-time staff to develop and maintain web standards. These standards are used to help web developers and browsers develop code that lives up to certain standards. In short, they write a manual that helps define whether our code is well written or poorly written. This is how we know if our markup (code) is awesome or not.

Markup Validator is a free service by W3C that helps in checking the validity of web documents. Most web documents are written using markup languages, such as HTML or XHTML. These languages are defined by technical specifications, which usually include machine-readable formal grammar (and terminology). The task of checking a document against these constraints is called verification, and that is what the markup validator does.

Validating Web documents is an important step which can dramatically help improving and ensuring their quality, and it can save a lot of time and money (read more on why validating matters). Validation is, however, neither a full quality check, nor is it strictly equivalent to checking for conformance to the specification.

This validator can process documents written in most markup languages. Supported document types include the HTML (through HTML 4.01) and XHTML (1.0 and 1.1) family, MathML, SMIL and SVG (1.0 and 1.1, including the mobile profiles). The Markup Validator can also validate Web documents written with an SGML or XML DTD, provided they use a proper document type declaration.

This validator is also An HTML validating system conforming to International Standard ISO/IEC 15445—HyperText Markup Language, and International Standard ISO 8879—Standard Generalized Markup Language (SGML) – which basically means that in addition to W3C recommendations, it can validate according to these ISO standards.

Fig: W3C checking

The screenshot is the process of checking W3C validator.

Why is it important to W3C valid your site?

To ensure that the site works properly on your browser as well as the other major browsers such as Internet Explorer, Google Chrome, etc, W3C validator is important. A website that contains many errors can suffer from many issues. Following are some of it:

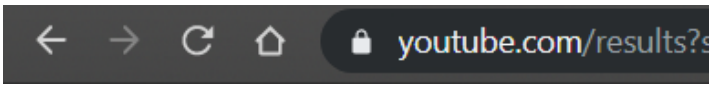
i. Not displaying properly or correctly.
ii. Not compatible with a major browsers.
iii. Loads elements slowly.
iv. Ranks badly in search engines due to incorrectness of code reading.

Secure Sockets Layer (SSL)

SSL is must in websites because it secures the transfer of data and has SEO benefits too. Having SSL certificates could earn your users trust. Using SSL can SSL and TLS are protocols for establishing authenticated and encrypted links between networked computers.

SSL/TLS works by binding the identities of entities such as websites and companies to cryptographic **key pairs** via digital documents known as X.509 certificates. Each key pair consists of a **private key** and a **public key**. The private key is kept secure, and the public key can be widely distributed via a certificate.

You can see ‘locked icon’ representing SSL certified.



The special mathematical relationship between the private and public keys in a pair mean that it is possible to use the public key to encrypt a message that can only be decrypted with the private key. Furthermore, the holder of the private key can use it to **sign** other digital documents (such as web pages), and anyone with the public key can verify this signature.

If the SSL/TLS certificate itself is signed by a **publicly trusted certificate authority (CA)**, such as **SSL.com**, the certificate will be implicitly trusted by client software such as web browsers and operating systems. Publicly trusted CAs have been approved by major software suppliers to validate identities that will be trusted on their platforms. A public CA’s validation and certificate issuance procedures are subject to regular, rigorous audits to maintain this trusted status.

SSL certificates bind together:

- A domain name, server name or hostname.
- An organizational identity (i.e. company name) and location.

How Does an SSL Certificate Work?

SSL Certificates use something called public key cryptography. This particular kind of cryptography harnesses the power of two keys which are long strings of randomly generated numbers. One is called a private key and one is called a public key. A public key is known to your server and available in the public domain. It can be used to encrypt any message. If Gaurav is sending a message to Saurav he will lock it with Saurav’s public key but the only way it can be decrypted is to unlock it with Saurav’s private key. Bob is the only one who has his private key so Saurav is the only one who can use this to unlock Gaurav’s message. If a hacker intercepts the message before Saurav unlocks it, all they will get is a cryptographic code that they cannot break, even with the power of a computer.

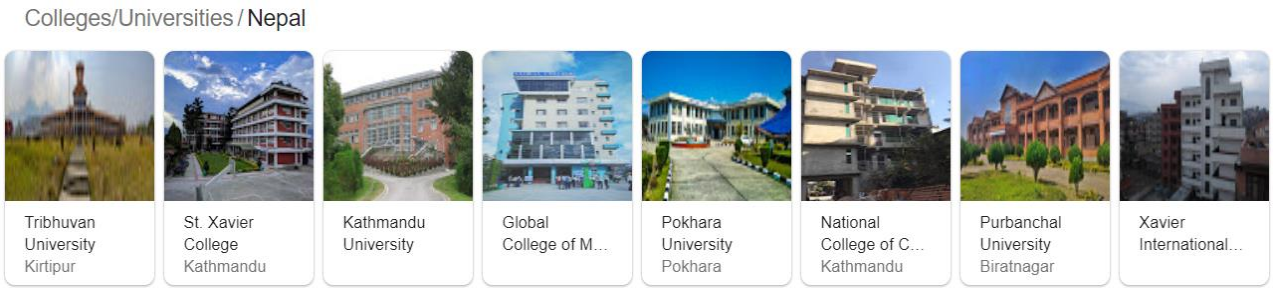
If we look at this in terms of a website, the communication is happening between a website and a server. Your website and server are Gaurav and Saurav.

SEO (Search Engine Optimization)

SEO is a process of improving the

Rankings

Rankings in SEO (Search Engine Optimization) refers to a website's position in the search engine results page. There are various ranking factors that influence whether



a website appears higher on the SERP based on the content relevance to the search term, or the quality of backlinks pointing to the page. Ranking in SEO (Search Engine Optimization) refers to the position of a website in the search engine result page. There are various ranking factors that impact whether a website appears more on the SERP depending on the content relevance of the search term.

As you can see in the screenshot, rakings of top colleges in Nepal.

Different search engines with different rankings

The ranking of a website or URL for a keyword or keyword combination varies from search engine to search engine. A domain can rank in the top 3 on Bing for a certain keyword, but may not even be on the first page of Google search results for the same keyword. Of course, the same can be said of all search engines - Bing, Google, Yahoo and every other search engine uses its own method of calculating rankings and therefore ranks websites differently.

Rankings can also vary when using different language or country versions of the same search engine, such as Google. A search analysis software like Search Matrix Suite is a good way to track and track ranking differences in search engines.

Good rankings are a good source of traffic

The purpose of good ranking in search results is to get more and more traffic from the organic search channel. The higher a page is in the results for the search query, the more likely the searcher will click on this result. This explains the direct relationship between high ranking and increasing traffic.

This relationship between ranking and clicks (and traffic) is strongest in the top 3 search results. However, with the inclusion of Google's Knowledge Graph data and the integration of Universal Search Elements (SERP features) such as videos, maps, and Google Shopping ads, the layout of search results pages is constantly changing. These developments could mean that the top 3 organic rankings are no longer the 3 best places on the SERP. This has been demonstrated in heat map and eye-tracking tests.

What can influence my rankings?

Here, Search engine optimization is a method for sustainably influences search engine rankings. Google and other search engines calculate their search results for keywords using their highly complex algorithms. The individual ranking factors and their weighting within the ranking calculation are well-guarded abstract property that belongs to the search engines and is not publicly impart.

The following factors can be factors to rankings:

- Number of backlinks.
- Sitemap and internal linking.
- Usage of keywords in text elements like Meta titles, Meta descriptions, text etc.
- Term optimization of content, based on comparison with other documents on the same topic (proof and relevant terms, topic/content clusters, WDF*IDF).
- URL structure.
- Trust assigned to the page.
- Page load time (site speed).
- Time on site and bounce rate (here: how long a visitor spends on the page before they return to the SERP).
- CTR in the SERPs, i.e. how often searchers click on the result.
- Presumably many other factors like page traffic, authorship, how up-to-date a page is. etc.

Discussing the capabilities and relationships between front-end and back-end website technologies and explaining how these relate to presentation and application layers.

We know that, Frontend and Backend are the two most popular terms used in the web industry, but their difference are quite indistinct. They are the two most fundamental parts of

software engineering because it plays a vital role in web development. As we already discussed about frontend and backend, we will be talking about their capabilities and relationships.

Role of Frontend and Backend

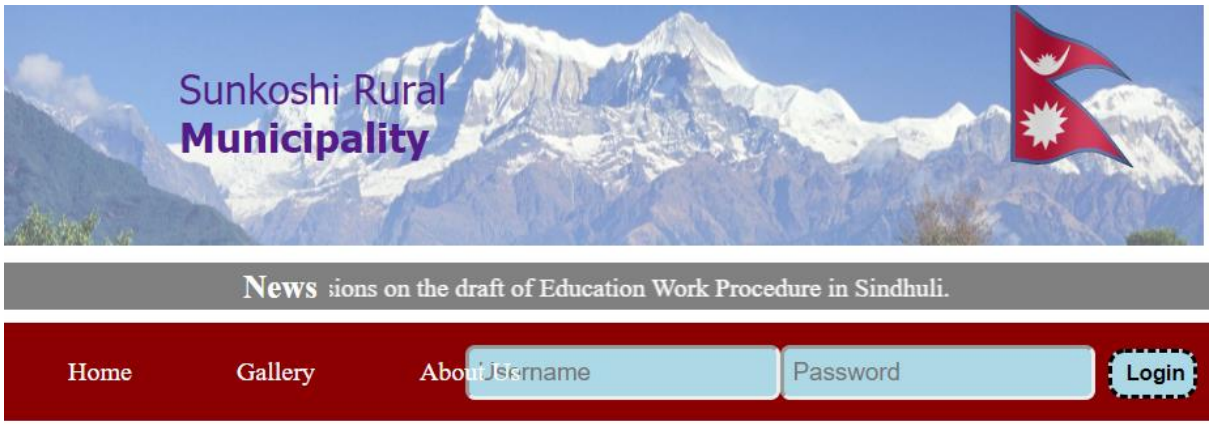
These both play a vital role in web development and although they have similar share of differences, they are alike two sides of the same coin. Frontend is all about the visual aspects of the website that a user can see and experience. On the opposite, everything that happens on the background can be attributed to the backend web development. It's more like an assistant for frontend web experience.

Differences between Frontend and Backend

Even these terms are widely used in web development, they have some differences:

S.N.	Front-End	Back-End
1.	Frontend also refers to the client-side of the application.	Backend also refers to the server side of the application
2.	It is the part of the website users can see and interact with.	It constitutes everything that happened behind the scenes.
3.	It basically include everything that attributes to the visual aspects of websites.	It generally includes a web server that communicates with a database to serve requests that the frontend presents.
4.	HTML, CSS, JavaScript includes here.	PHP, Python, Java, Ruby, etc includes here.

This is my website's screenshot of front-end where user see and interacts with it. HTML, CSS are used here.



Examples of Frontend and Backend

Concepts and components that focus on the front end of a system includes:

- Design and markup languages like HTML, CSS and JavaScript.
- Search engine optimization (SEO).
- Usability and accessibility testing.
- Graphic design and image editing tools.
- Web performance and browser compatibility.

Inversely, those that focuses on the back end of a system includes:

- Programming and scripting languages like PHP, Python and C#.
- Automated testing frameworks.

- Network scalability and availability.
- Database management and data transformation.
- Cybersecurity and data backup practices.

Relation of Frontend and Backend with different layers.

Front-end and back-ends mean separation of concerns between presentation layer, application layer and database layer as well as application in front-end and back-end has two layers i.e. application and database layer. Three different layers of front-end and back-end are described below.

Presentation Layer The presentation layer is known as the front-end layer that contains a user interface and graphical user interface that can be accessed through web browsers and web applications that describe content and information useful to the end user. Similarly, this application layer is built using web technologies such as HTML, CSS and JavaScript and can also be built using the framework and communicate with other layers by API calls. Application Layer The application layer consists of functional business logic that runs the core capabilities of an application, written mostly in Java, .NET, C #, Python, C ++, etc. Database layers are database / data storage systems and data access layers as part of the database layer and examples of these systems include MySQL, Oracle, PostgreSQL, MongoDB, Microsoft SQLServer, etc. and access data via application layer API calls. The presentation layer is deployed on most desktops, tablets and phones to use a web server either through a web browser or through a web-based application, with the application layer mostly hosted in the cloud or in a dedicated workstation. The application is done which depends on the required complexity and processing power. There are various benefits of using a three-layer architecture such as speed of development, scalability, performance and availability as well as it helps improve development efficiency by allowing the team to focus on core competencies.

Note: There are also many people who work and understand both the front-end and back-end. So, those people are often called Full Stack Developers.

Discuss the differences between online website creation tools and custom built sites with regards to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI)

Most websites are developed with a content management system (CMS) which simply allows you to easily and quickly update their sites without any technical support. Online website creation tools can also be called as Template Site.

Using Templates

Here, a website templates is a pre-designed webpage, or set of HTML and CSS webpages where anyone can use to 'plug-in' their own header, text, photos, background, etc to create a website. It allows you to setup a website without hiring a professional web developer or designers. The most common way to start a website would be to set up a WordPress site and apply a template. These websites provide a broad range of different types of website templates available for purchase. Not all of the themes/templates out there are 100% responsive.

There are also many services like Wix, WordPress, Weebly, GoDaddy or Squarespace that also provide a theme and allow you to edit within the given frame of the website. They can provide a website pretty fast and easily. There are also limitations on the options that are available for more advanced features such as E-Commerce functions.

Keep in mind that templates and themes comes with built-in feature so that the user can have more customizability but this ultimately weighs down the website in terms of load speed and SEO performance. Besides that, you can pay a web development company to fix it, but that's like asking a certified ford mechanic to take a look at your Toyota since the site most likely won't be written in the same code format.

Pros/Cons of using templates:

1. Lots of other businesses could be using the same template which means your site won't stand/value out as much.
2. You are limited on how much you can customize the site. What you see is what you get.
3. May not 100% work on all devices i.e. it may not be responsive.

4. Some templates are not built to be search engine friendly. They need to be customized to fit your business.
5. Any custom or added technologies are not possible to be installed as templates run on a structured system.
6. Templates are great for low budget. There is a cheaper upfront cost.
7. Templates have a shorter development time than that of custom built site
8. Template sites will mostly likely not have as good of a support system as a custom built site.

Custom Built Websites

Custom built websites include a team behind your business. It starts with a creative method to understand who your audience is, who you want to reach, how you want to work / need a website and how you want to be seen on the Internet.

We already discussed that SEO plays a big role in the success of any website. There are rules to follow when creating a site and not all subjects are able to follow these rules. This is why custom built websites are so much better, the developer is able to optimize the site during the build so that Google and other search engines are allowed to read every page in order to give more relevant search results to the end user. So to receive. Naturally a better ranking on the system. This is beyond keyword research.

The reason that custom built websites take longer than a theme is because everything is being built from the starting phase. This will surely save your time in the long period of time as it is very easy to optimize/customize the site when it is built from the ground up to ensure that your site can do what you want it to do. Custom designs allow your site to be responsive across all devices (such as phones, tablets, laptops, etc.) and browsers.

Pros/Cons of using Custom Build Sites:

1. The website will have a completely unique design that will incorporate the company's branding and will be based around the needs of your requirement.
2. The site is tested to work across all browsers.
3. The site will be built to be search engine friendly. I.e. SEO.
4. A customized website is more expensive than a template.

5. The process to create a custom site takes longer than a template site.
6. The site will grow with your business. If you know how you will need your site to operate in the future, a professional web developer will install technologies that will work with those ideas.
7. The company that built your site is there for support if you need it.

Differences between Template Websites and Custom Build Websites:

Here are the differences between them:

S.N	Template Website	Custom Built Website
1.	The development time is generally shorter.	The development time is normally 6-8 weeks.
2.	Templates have a cheaper upfront cost.	Custom sites are comparatively more expensive than templates.
3.	Most templates are not SEO friendly.	Optimized for search engines during development.
4.	Other business may be using the same templates, so it won't look good.	Unique design based on the needs of the business
5.	Customization is limited. What you see at first is what you get.	You can customized anytime as your needs.
6.	Not 100% guaranteed to be responsive on all the devices.	100% responsive on all the devices.
7.	Doesn't include a reliable support system	The company that built your site is there for support.

So, these are the differences that differ templates based websites and custom build sites.

How to check if the site is template based or custom based:

It's very easy to see if a website is a template website or custom-built website. Firstly, view the site's source, here's how to access the code:

- **In Google Chrome:** Open the site and go to the “Tools” menu, and select “view source”; alternatively, hit Ctrl + U
- **In Internet Explorer, Mozilla Firefox, or Apple Safari:** Right-click on the website and choose “View source”

You’ll see a page of source coding, which may be pretty intimidating if this is unfamiliar territory. Toward the top of the page, in the paragraph that begins <head>, scan through the code and look for the “generator” meta tag to see which Content Management System is being used.

Examples:

- A website built on the WordPress platform may contain this: <meta name=”generator” content=”WordPress 3.8.1″ />.
- A website built on the Joomla platform may contain this: <meta name=”generator” content=”Joomla! 1.5 – Open Source Content Management” />.

Conclusion

Hence, I have prepared a report that identifies purpose and types of DNS along with how domain names are organized and managed. Also, I included the purpose and relationships between communication protocols, server hardware, operating systems and web server software in regards to web designing, publishing and accessing a website. This part also contains the influence of search engines on website performance along with website crawling, indexing and ranking and provide.

I also evaluated the impact of common web development technologies and frameworks with regards to website design, functionality and management. Also, explained the capabilities and relationships between front-end and back-end website technologies along with their relation to presentation and application layers. At last, the differences between online website creation and custom built sites were also shown.

Part 2

In the second part the company wants you to develop a complete website following the overall process of software development from the requirements specification to development and testing. The project is for a Sunkoshi Rural Municipality which is in Sindhupalchowk district. They want an online system of data collection along with the functionalities like view the records, edit and delete as per need. Data collection can be categorized as: personal, organizational, industrial etc. Personal details should include name, address, profession, date of birth etc. Organizational further can be broken down into government owned, private, educational, hospitals etc. Industrial may include the local level mini and micro industries. Record entry form, its view, edit, delete functionalities should be different for different categories.

1. From the requirements above create a proposal document for the multipage website with fidelity wireframes with full set of client and user requirements. This must be convincing to them and should contain exactly what they want. 2. After the requirements specification phase, develop a website with above all functionalities. The website must contain at least 10 web pages. Should have the CRUD functionality i.e. create, retrieve, update and delete.

3. Finally, write a report comparing the multipage website created to the design document and critically evaluate the design and development process against your design document analyzing any technical challenges.

4. Create a suitable test plan to test your developed website identifying key performance areas and use it to review the overall functionality and performance of the website. Evaluate the Quality Assurance (QA) process and review how it was implemented during your design and development stages and finally critically evaluate the result of your test plan and include the review of the overall success of your website, use this to explain any areas of success and provide justified recommendations for areas that require improvement.

Introduction

In this part II, I am developing a complete website following the overall process of software development from the requirements specification to development and testing. This project is for a Sunkoshi Rural Municipality which is in Sindhupalchowk district. They want an online system of data collection along with the functionalities like view the records, edit and delete as

per need. Data collection are categorized as: personal, organizational, industrial etc. Personal details includes name, address, profession, date of birth etc. Organizational are broken down into government, educational, hospitals etc. Record entry form, its view, edit, delete functionalities are different for different categories.

From the requirements above, I have created a fidelity wireframes with full set of client and user requirements. After this, this website contains more than 10 pages and have CRUD functionality. At last, I have created test plan to test my developed website identifying key performance areas and use it to review the overall functionality and performance of the website.

Creating a design document for a branded, multipage website with wireframes and full set of client and user requirements

Proposal Document

Here, I have developed a multipage website for a Sunkoshi Rural Municipality which is in Sindhupalchowk district. In this website, an online system of data collection is included where different people and organization give and receive information about Sunkoshi Rural Municipality. I have created most of the pages with view, edit, add, and delete functionalities.

Objectives

1. To provide information about the district
2. To analyze the number of people and organizations

Scope

Here, at the end of the project users will be able to perform many task and also rural people will be able to use internet and other stuffs. Different people from Sunkoshi will be able to create their account and provide their personal details which will be helping them to analyze the overall data. Different organizations like industrial, educational, governmental etc will be recorded here for a reason.

Timeframe

Time is something we have to manage on time. For making this website, I have allocated some period of time given below:

	Description of work	Start and End date

Phase 1:	Discovery Phase	1st August – 21st August
Phase 2:	Design Phase	22nd August – 3rd October
Phase 3:	Development Period	4th October – 25th October
Phase 4:	Modification and Testing Period	26th October – 9th November
Phase 5:	Launch and Maintenance	Approx. 2 weeks.

Phase 1: Discovery

Discovery is how we turn your problems into a plan.

It starts with a questionnaire covering your organization, products and services, business goals, customer demographics, competition and more.

Following an investigative kick-off conference call, we take a deep dive into your analytics data to identify and investigate your business, brand, industry competitors, any current sites you may have, and your problems. In some cases, we conduct a technical audit of your existing site to understand technical debt.

We then analyze our findings and document the plan to solve your problems, and work with the plan to make sure it meets your specific needs. It outlines the organization of your site as to how users will navigate it, the order of page content, and when users will need to manipulate the content.

We use this document to identify key pages, page types, content requirements, functionality, and technical requirements, the scope of the project used to provide a definitive bid for the design.

Phase 2: Design

The design is how we transform the plan into a mockup.

You provide available brand assets - which may include logos, colors, directions, photography, etc. - and we source additional assets such as web fonts, icons, and stock photography.

Using planning documentation including visitor browser and device data, we establish a responsive layout grid, typography, and base style guide with title and text styles, buttons, blocks, and more.

We then design user interface elements such as site navigation, footers, tab bars, cover images, call-to-action areas, cards, posts, lists, form fields, and other repeatable visual elements for desktop and mobile.

With these elements defined, we begin the design search of the main pages, iterating the layout until we have mockups that meet the planning requirements. We then work with you to refine the main page mockup to meet your specific needs.

We use completed mockups to establish technical work and provide a definitive bid for development.

Phase 3: Development

Development is how we turn a mockup into a fully functional website.

First we set up a version controlled development environment and then, following full designs, we build a blazing fast and SEO friendly website using modern tools and uniquely crafted code. The backend is easy to manage, providing a custom, tailor-made delivery. , High performance WordPress theme.

We use industry leading plugins when it makes sense - for example, using WPForms for contact forms - and specifically custom calendars for you, including event calendars, API integrations, and more Use custom plugins for e.g. No matter how rich and complex a website is, a non-technical editor can easily maintain it. We provide video tutorials for WordPress training, detailed editing notes explaining custom-developed features, and a phone-based walkthrough of your website.

After initial development, you will review a fully functional site in a staged environment where you will update features and content.

Once approved, we will transfer the site to your live web server. We can also provide hosting recommendations to ensure that your site keeps on burning after launch.

Phase 4: Testing and Modification

Once the site has been developed, next phase is testing and modification where the client gets opportunity to share the site with the necessary stakeholders for review and feedback in

password-protection location. Testing is also conducted to ensure the site is optimized across multiple browsers, screens and mobile devices and ensure the site is performing as it should prior to launch.

Phase 5: Launch and Maintenance

Once the site is all set, it is launched simply and quickly or can be more involved depending on what is involved. Right after launching, there needs to be ongoing upkeep in terms of the hosting and security. We can keep improving the site based on real user feedback and requests.

Project Budget

Here, budget is a main thing in any sector. Cost of building a website cost:

Factors	Cost (in NRS)
Setup	16,000
Design and Building	5,00,000
Content Creation	50,000
Training	60,000
Maintenance	50,000
Total	6,76,000

Stakeholders

As a marketer or business owner, part of your core responsibilities is to meet the requirements imposed by stakeholders to whom your business has obligations. These stakeholders can be involved in many different aspects of your day to day affairs. Following maybe our stakeholders:

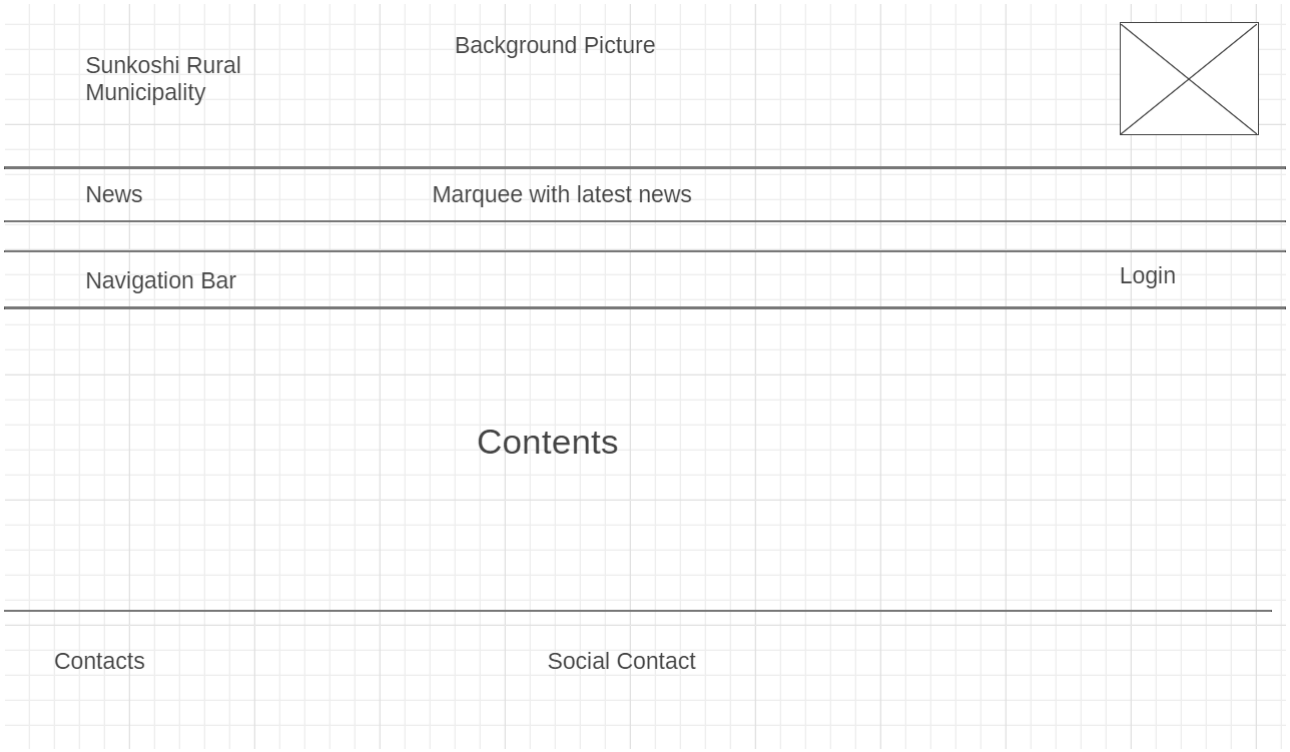
1. People (High interest, Low power)
2. Employees (High interest, High power)
3. Government (Low interest, High power)

Wireframe

Wireframe is a layout of a webpage that describes what interface elements are contained on major pages. It is a critical part of the interaction design process. The role of a wireframe is to give a visual understanding of a page early in a project to get stakeholder and project team approval before the creative phase gets under way. It can also be useful to create global and secondary navigation to make sure that the terminology and structure used for the site meets user expectations. Wireframe for my webpages are as follows:

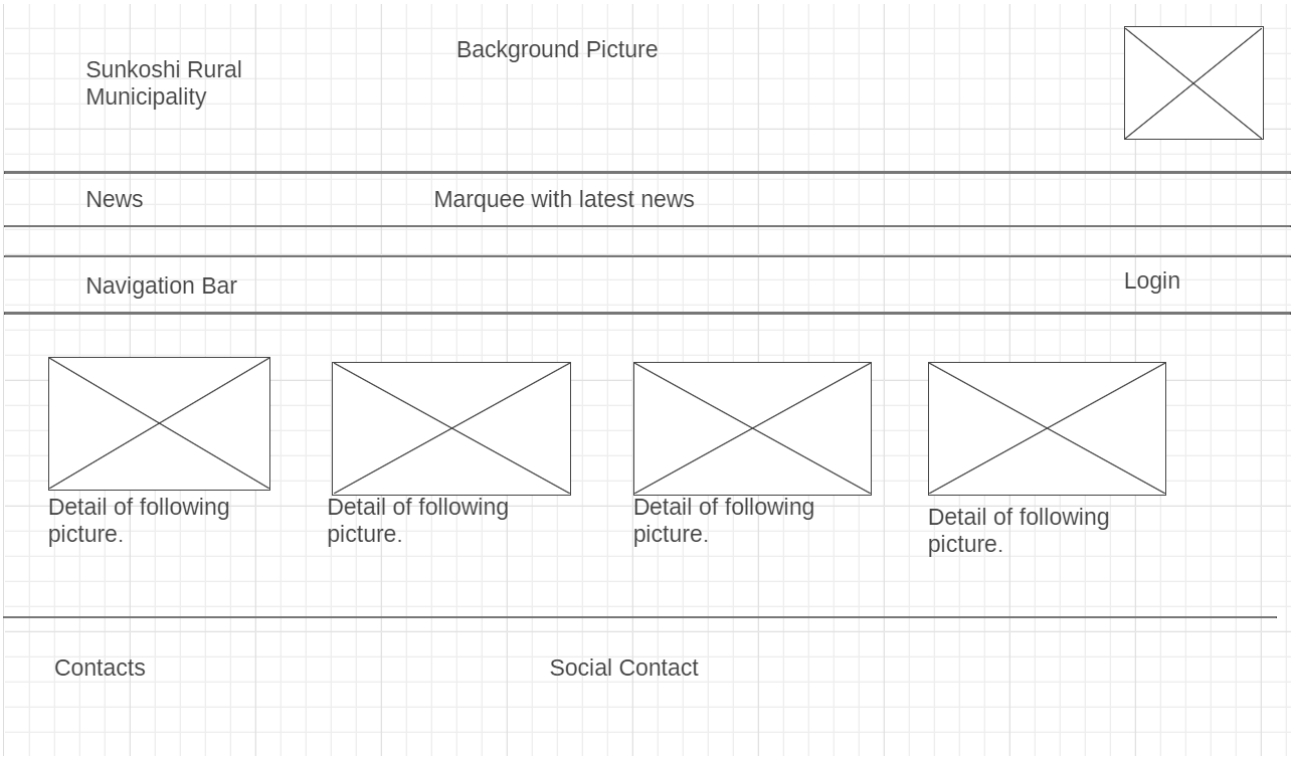
a) Home Page

This is homepage where user can login through and see contents of Sunkoshi Rural Municipality.



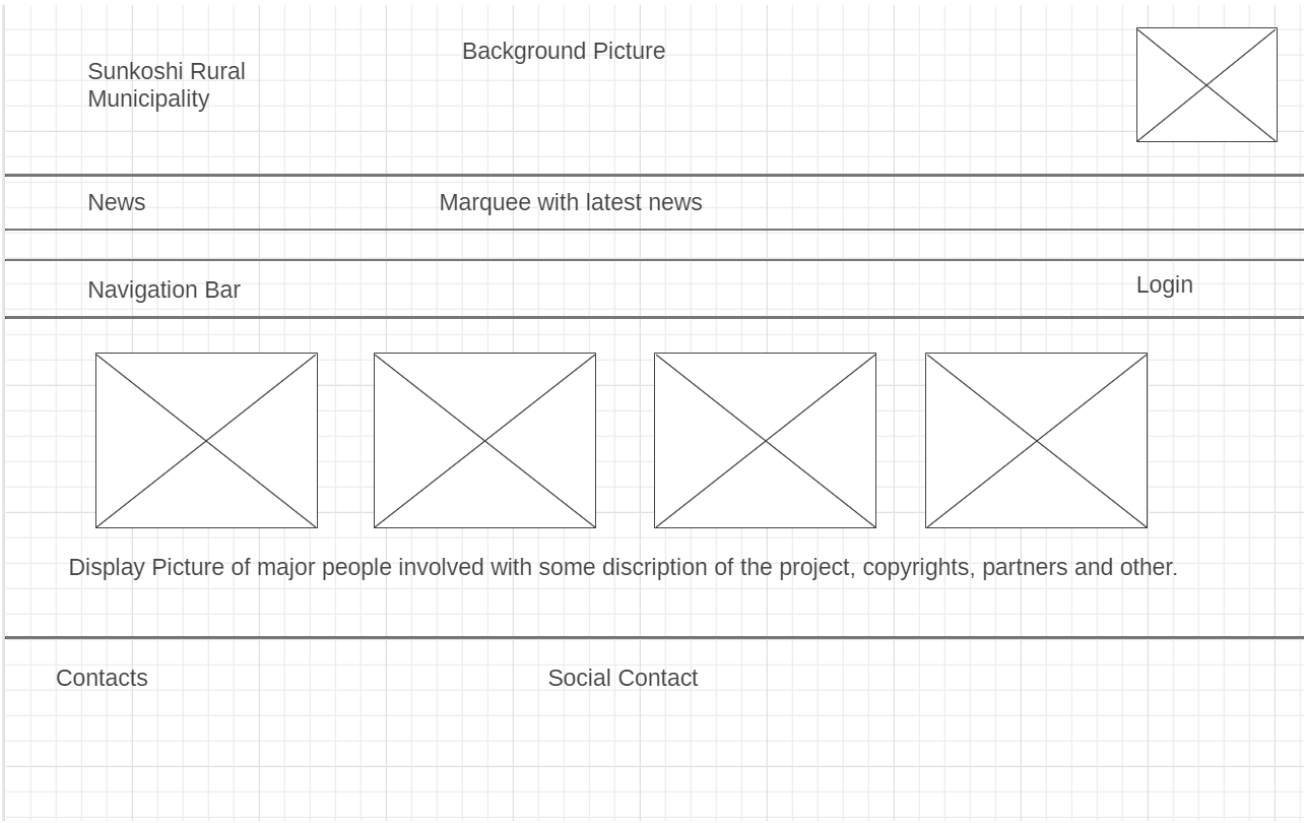
b) Gallery Page

We can find many pictures and videos of Sunkoshi in this page and take various information.

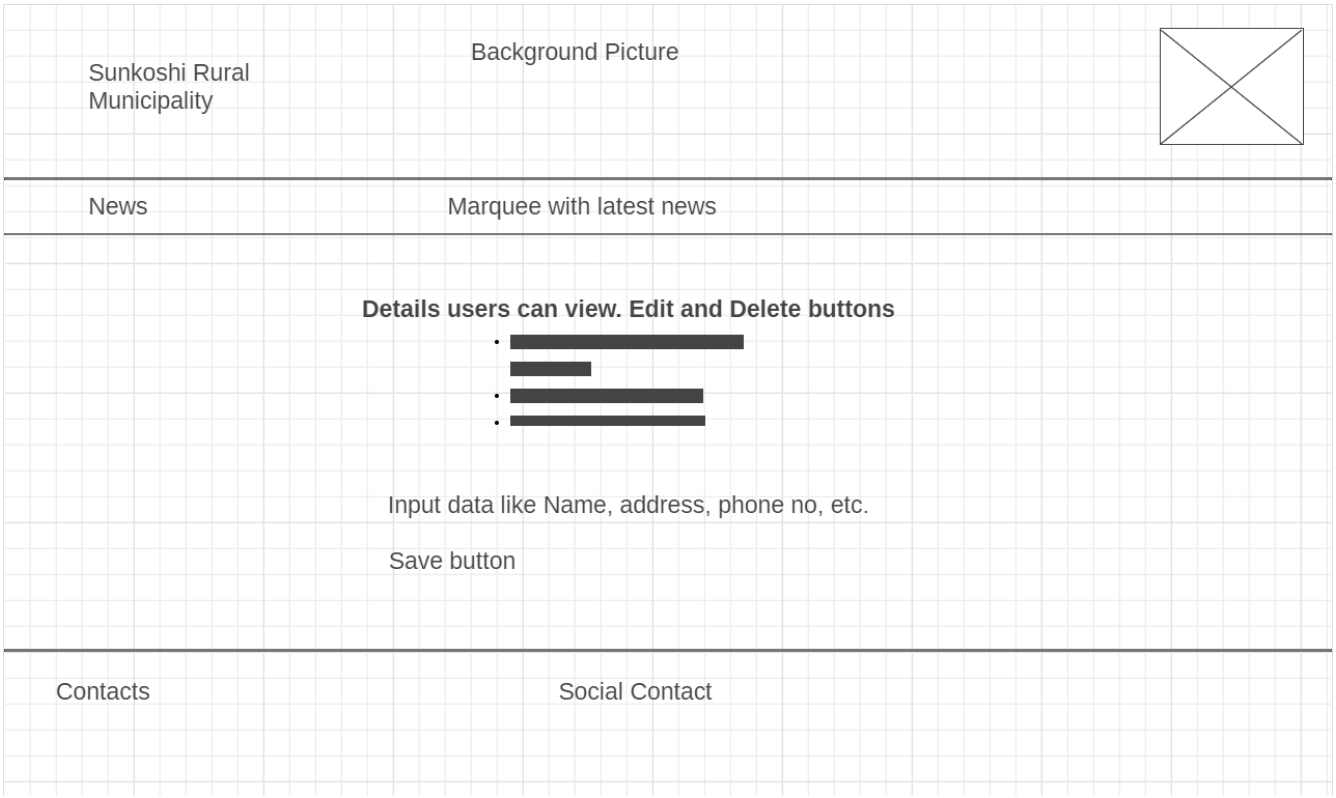


c) About Us Page

In this page, all the required information related with our website like founder, writer, editor etc can be found.



d) Homepage after login



e) Details Page

Here, user can give their respective details and can edit and delete too.

Header File:

gallery.php | aboutus.php | crudindex.php | crudphp.php

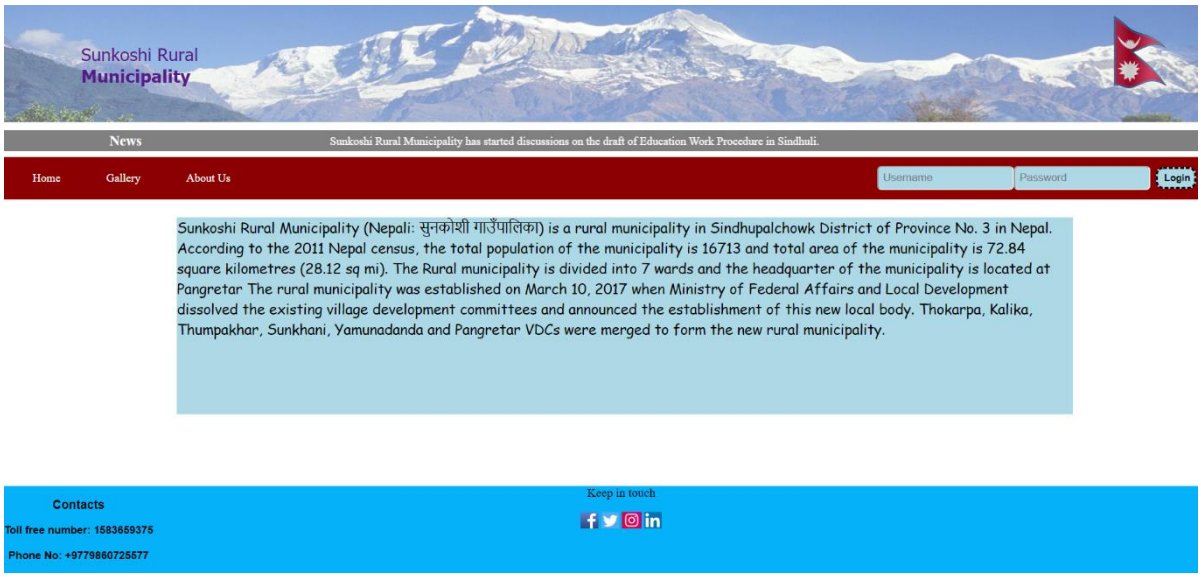
```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
    <body >
        <style type="text/css">

        .bgimage #bg{

        margin: -8px;
        opacity: 0.6;
        }
        #flag{
            width: 80px;
            margin-top: -150px;
            margin-right: 50px;
            position: relative;
            float: right;
        }
        .head{
            position: absolute;
            margin-top: -150px;
            overflow: hidden;
            margin-right: 200px;
            font-size: 25px;
            color: tomato;
            font-family: Tahoma, Verdana, Segoe, sans-serif;
            margin-left: 100px;
            }


        #bg {
            width: 100%;
            min-height: 150px;
        }
        .head a{
            text-decoration: none;
        }


        .thisismarquee{
            margin-top: -50px;
            margin-left: 200px;
            text-align: center;
```

```
            }
        .mar{
            margin-left: -8px;
            margin-top: -30px;
            margin-right: -8px;
            background-color: gray;
            color: white;
        }
        .news {
            width: 280px;
            text-align: center;
            margin-left: 30px;
            font-size: 20px;

        }
        .news h4{
            line-height: 30px;
        }

        </style>


        <div class="topnav">
        <div class="bgimage">
            <img id="bg" src="BG.jpg">
        </div>
        <div class="flag">
            <img id="flag" src="nepalflag.gif">
        </div><br>
        <div class="head">
            <a href="home.php"><p>Sunkoshi Rural <br><b>Municipality</b></p></a>
        </div>
    </div>
    <div class="mar">
            <div class="news">
                <h4>News</h4>
            </div>
            <div class="thisismarquee">
                <marquee>Sunkoshi Rural Municipality has started discussions on the draft of Education Work Procedure in Sindhuli. </marquee>
            </div>

        </div>


</body>
</html>
```

Server file:

```
<?php
session_start();

// initializing variables
$username = "";
$email    = "";
$errors = array();

// connect to the database
$db = mysqli_connect('localhost', 'root','','multi_login');

// REGISTER USER
if (isset($_POST['reg_user'])) {
  // receive all input values from the form
  $username = mysqli_real_escape_string($db, $_POST['username']);
  $email = mysqli_real_escape_string($db, $_POST['email']);
  $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
  $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

  // form validation: ensure that the form is correctly filled ...
  // by adding (array_push()) corresponding error unto $errors array
  if (empty($username)) { array_push($errors, "Username is required"); }
  if (empty($email)) { array_push($errors, "Email is required"); }
  if (empty($password_1)) { array_push($errors, "Password is required"); }
  if ($password_1 != $password_2) {
  array_push($errors, "The two passwords do not match");
  }

  // first check the database to make sure
  // a user does not already exist with the same username and/or email
  $user_check_query = "SELECT * FROM users WHERE username='$username' OR email='$email' LIMIT 1";
  $result = mysqli_query($db, $user_check_query);
  $user = mysqli_fetch_assoc($result);

  if ($user) { // if user exists
    if ($user['username'] === $username) {
      array_push($errors, "Username already exists");
    }
```

Homepage



Code for it:

```php
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
    <title>Sunkoshi Rural Municipality</title>
    <meta charset="utf-8">

    <link rel="stylesheet" type="text/css" href="stylehomee.css">
    <link rel="icon" type="image/jpg" href="icon.ico">



</head>
<body>

    <?php include('header.php'); ?>


    <?php include ('navigationbar.php'); ?>



<br><br><br><br>
    <div class="contents">
        <p>Sunkoshi Rural Municipality (Nepali: सुनकोशी गाउँपालिका) is a rural municipality in Sindhupalchowk District of Province No. 3 in Nepal. According to the 2011 Nepal census, the total population of the municipality is 16713 and total area of the municipality is 72.84 square kilometres (28.12 sq mi). The Rural municipality is divided into 7 wards and the headquarter of the municipality is located at Pangretar

        The rural municipality was established on March 10, 2017 when Ministry of Federal Affairs and Local Development dissolved the existing village development committees and announced the establishment of this new local body.

        Thokarpa, Kalika, Thumpakhar, Sunkhani, Yamunadanda and Pangretar VDCs were merged to form the new rural municipality. </p>

        <br><br><br>
    </div>

    <?php include ('footer.php'); ?>
</body>
</html>
```

Gallery:

Code:

gallery.php

```php
<?php include('server.php'); ?>
<!DOCTYPE html>
<html>
<head>
    <title>Gallery</title>
</head>
    <link rel="stylesheet" type="text/css" href="stylehomee.css">
    <link rel="icon" type="image/jpg" href="icon.ico">
    <style>
        .img {
            float: center;
            margin-left: 150px;
            width: 2000px;
        }

    </style>
<body>

    <?php include('header.php'); ?>
    <?php include('navigationbar.php'); ?>


    <div class="img">
        <img src="helpinghand1.jfif">
    <img src="helpinghand2.jfif">
    <img src="helpinghand3.jfif">
    <img src="helpinghand4.jfif">
    <img src="helpinghand5.jfif">
    </div>

    <?php include('footer.php'); ?>

</body>
</html>
```

Login Code:

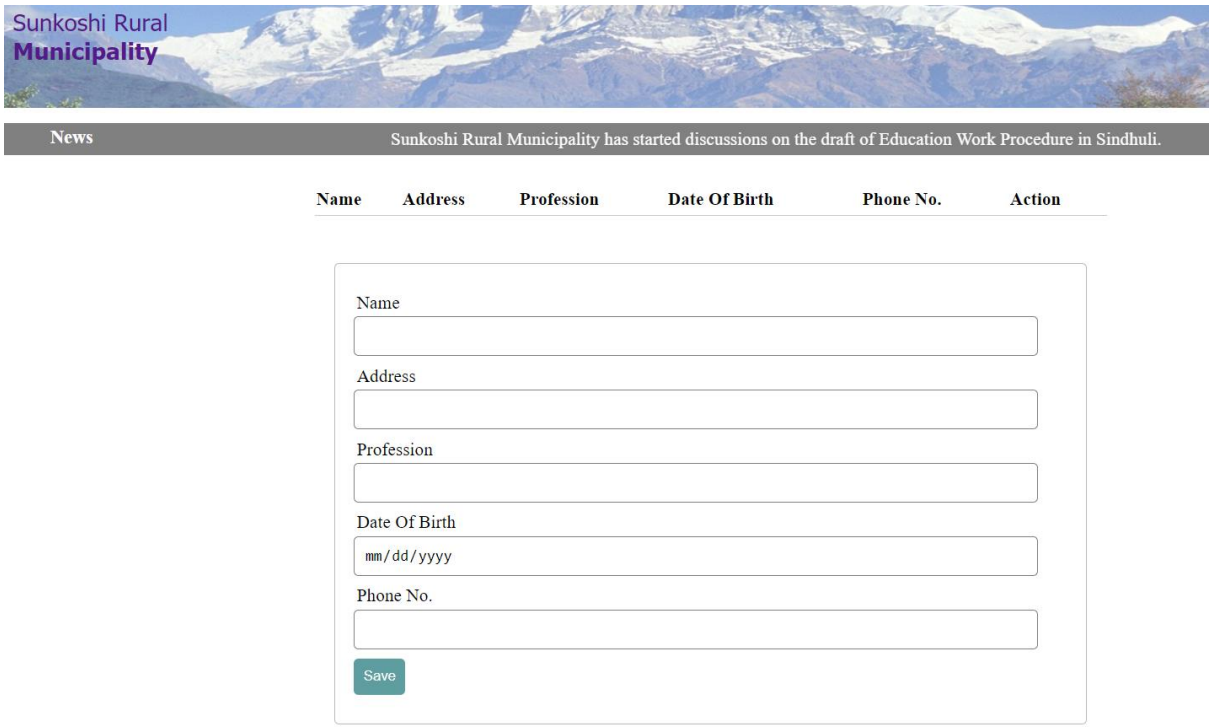
```php
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
  <title>Sunkoshi Rural Municipality</title>

  <link rel="icon" type="image/jpg" href="icon.ico">
  <link rel="stylesheet" type="text/css" href="styleph.css">

</head>
<body>
 <div class="topnav">
    <div class="bgimage">
      <img id="bg" src="BG.jpg">
      <p><img id="flag" src="nepalflag.gif"></p>
    </div>
  </div>
  <div class="Nav">
        <a href="home.php"><button id="backtohome" type="button">Back To Home</button></a>


      </div>
  <div class="formm">
    <div  class="header">
    <h2>Login</h2>
  </div>

  <form id="form" method="post" action="login.php">
    <?php include('errors.php'); ?>
    <div class="input-group">
      <label>Username</label>
      <input type="text" name="username" >
    </div>
    <div class="input-group">
      <label>Password</label>
      <input type="password" name="password">
    </div>
    <div class="input-group">
      <button type="submit" class="btn" name="login_user">Login</button>
    </div>
    <p>
      Not yet a member? <a href="register.php">Sign up</a>
    </p>
  </form>
  </div>

</body>
</html>
```

Login Homepage:

Code:

```php
<?php
  session_start();

  if (!isset($_SESSION['username'])) {
    $_SESSION['msg'] = "You must log in first";
    header('location: home.php');
  }
  if (isset($_GET['logout'])) {
    session_destroy();
    unset($_SESSION['username']);
    header("location: home.php");
  }
 ?>
 <!DOCTYPE html>
 <html>
 <head>
   <title>Home</title>
   <link rel="stylesheet" type="text/css" href="stylephpp.css">
   <link rel="icon" type="image/jpg" href="icon.ico">
 </head>
 <body>
 <?php include('header.php'); ?>
 <div class="header">
   <h2>Home Page</h2>
 </div>
 <div class="content">
     <!-- notification message -->
     <?php if (isset($_SESSION['success'])) : ?>
       <div class="error success" >
         <h3>
           <?php
             echo $_SESSION['success'];
             unset($_SESSION['success']);
           ?>
         </h3>
       </div>
     <?php endif ?>

   <!-- logged in user information -->
   <?php  if (isset($_SESSION['username'])) : ?>
     <p>  Welcome <strong><?php echo $_SESSION['username']; ?></strong></p>
     <br>
   <?php endif ?>
  <div class="list">
    <ol style="line-height: 50px;">
      <li><a href="personalhome.php">Personal Details</a></li>



      <div class="dropdown">
        <li>
          Organization Details
        </li>

  <div class="dropdown-content">
    <a href="industryhome.php">Industry Detail</a>
    <a href="educationhome.php">Education Detail</a>

  </div>
</div>
      <div class = "dropdown">
        <li>Government Organization</a></li>

        <div class = "dropdown-content">
          <a href="#">Office Organization</a>
          <a href="#">Police Departments</a>
          <a href="#">Hospital Departments</a>
        </div>
      </div>

  </div>



    <br><br><br>
    <p> <a href="indexxx.php?logout='1'" style="color: red;">logout</a> </p>
</div>

<?php include ('footer.php'); ?>

</body>
</html>
```

Personal Details:



Code:

1.

gallery.php | aboutus.php | crudindex.php | crudphp.php | crudstyles.c

```php
<?php  include('crudphp.php'); ?>
    <?php
    if (isset($_GET['edit'])) {
        $id = $_GET['edit'];
        $update = true;
        $record = mysqli_query($db, "SELECT * FROM info WHERE id=$id");

        if (@count($record) == 1 ) {
            $n = mysqli_fetch_array($record);
            $name = $n['name'];
            $address = $n['address'];
            $profession = $n['profession'];
            $dob = $n['dob'];
            $phone = $n['phone'];
        }
    }
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="crudstyles.css">
    <link rel="icon" type="image/jpg" href="icon.ico">
    <title>Personal Details</title>

    <link rel="stylesheet" type="image/jpg" href="flag.icon">
</head>
<body>
    <?php include('header.php'); ?>
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
        ?>
    </div>
<?php endif ?>
    <?php $results = mysqli_query($db, "SELECT * FROM info"); ?>

    <table>
    <thead>
        <tr>
            <th>Name</th>
            <th>Address</th>
            <th>Profession</th>
            <th>Date Of Birth</th>
            <th>Phone No.</th>
```

```
            <th>Profession</th>
            <th>Date Of Birth</th>
            <th>Phone No.</th>

            <th colspan="2">Action</th>
        </tr>
    </thead>

    <?php while ($row = mysqli_fetch_array($results)) { ?>
        <tr>
            <td><?php echo $row['name']; ?></td>
            <td><?php echo $row['address']; ?></td>
            <td><?php echo $row['profession']; ?></td>
            <td><?php echo $row['dob']; ?></td>
            <td><?php echo $row['phone']; ?></td>

            <td>
                <a href="personalhome.php?edit=<?php echo $row['id']; ?>" class="edit_btn" >Edit</a>
            </td>
            <td>
                <a href="crudphp.php?del=<?php echo $row['id']; ?>" class="del_btn">Delete</a>
            </td>
        </tr>
    <?php } ?>
</table>

    <form method="post" action="crudphp.php" >
        <input type="hidden" name="id" value="<?php echo $id; ?>">
        <div class="input-group">
            <label>Name</label>
            <input type="text" name="name" value="<?php echo $name; ?>">
        </div>
        <div class="input-group">
            <label>Address</label>
            <input type="text" name="address" value="<?php echo $address; ?>">
        </div>
        <div class="input-group">
            <label>Profession</label>
            <input type="text" name="profession" value="<?php echo $profession; ?>">
        </div>
        <div class="input-group">
            <label>Date Of Birth</label>
            <input type="Date" name="dob" value="<?php echo $dob; ?>">
        </div>
```

```
        <div class="input-group">
            <label>Date Of Birth</label>
            <input type="Date" name="dob" value="<?php echo $dob; ?>">
        </div>
        <div class="input-group">
            <label>Phone No.</label>
            <input type="text" name="phone" value="<?php echo $phone; ?>">

        </div>
        <div class="input-group">
            <?php if ($update == true): ?>
   <button class="btn" type="submit" name="update" style="background: #556B2F;" >update</button>
?php else: ?>
   <button class="btn" type="submit" name="save" >Save</button>
?php endif ?>
        </div>
   </form>


?php include ('footer.php'); ?>


/body>
/html>
```

```
    <form method="post" action="crudphp.php" >
        <input type="hidden" name="id" value="<?php echo $id; ?>">
        <div class="input-group">
            <label>Name</label>
            <input type="text" name="name" value="<?php echo $name; ?>">
        </div>
        <div class="input-group">
            <label>Address</label>
            <input type="text" name="address" value="<?php echo $address; ?>">
        </div>
        <div class="input-group">
            <label>Phone No.</label>
            <input type="text" name="phone" value="<?php echo $phone; ?>">

        </div>
        <div class="input-group">
            <?php if ($update == true): ?>
    <button class="btn" type="submit" name="update" style="background: #556B2F;" >update</button>
<?php else: ?>
    <button class="btn" type="submit" name="save" >Save</button>
<?php endif ?>
        </div>
    </form>



</body>
</html>
```

2.

gallery.php | aboutus.php | crudindex.php | crudphp.php | indexoo

```
<?php  include('crudphp.php'); ?>
    <?php
    if (isset($_GET['edit'])) {
        $id = $_GET['edit'];
        $update = true;
        $record = mysqli_query($db, "SELECT * FROM info WHERE id=$id");

        if (@count($record) == 1 ) {
            $n = mysqli_fetch_array($record);
            $name = $n['name'];
            $address = $n['address'];
            $phone = $n['phone'];
        }
    }
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="crudstyles.css">
    <title>Personal Details</title>

    <link rel="stylesheet" type="image/jpg" href="flag.icon">
</head>
<body>
    <div class="">
        <div class="topnav">
        <div class="bgimage">
            <img id="bg" src="BG.jpg">
            <p><img id="flag" src="nepalflag.gif"></p>

        </div>
    </div>

    </div>
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
```

```php
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
        ?>
    </div>
<?php endif ?>
    <?php $results = mysqli_query($db, "SELECT * FROM info"); ?>

    <table>
    <thead>
        <tr>
            <th>Name</th>
            <th>Address</th>
            <th>Phone No.</th>
            <th colspan="2">Action</th>
        </tr>
    </thead>

    <?php while ($row = mysqli_fetch_array($results)) { ?>
        <tr>
            <td><?php echo $row['name']; ?></td>
            <td><?php echo $row['address']; ?></td>
            <td><?php echo $row['phone']; ?></td>
            <td>
                <a href="personalhome.php?edit=<?php echo $row['id']; ?>" class="edit_btn" >Edit</a>
            </td>
            <td>
                <a href="crudphp.php?del=<?php echo $row['id']; ?>" class="del_btn">Delete</a>
            </td>
        </tr>
    <?php } ?>
</table>
```

3.

gallery.php | aboutus.php | crudindex.php | crudphp.php | indexxx.php | personalhome.php | login.php | erro

```php
<?php
    session_start();
    $db = mysqli_connect('localhost', 'root', '', 'crud');

    // initialize variables
    $name = "";
    $address = "";
    $id = "";
    $phone = "";
    $profession = "";
    $dob="";
    $update = false;

    if (isset($_POST['save'])) {
        $name = $_POST['name'];
        $address = $_POST['address'];
        $profession = $_POST['profession'];
        $dob = $_POST['dob'];
        $phone = $_POST['phone'];

        mysqli_query($db, "INSERT INTO info (name, address,profession,dob,phone) VALUES ('$name', '$address','$profession',$dob,'$
            phone')");
        $_SESSION['message'] = "Saved successfully :)";
        header('location: personalhome.php');
    }

        if (isset($_POST['update'])) {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $address = $_POST['address'];
    $profession = $_POST['profession'];
    $dob = $_POST['dob'];
    $phone = $_POST['phone'];
```

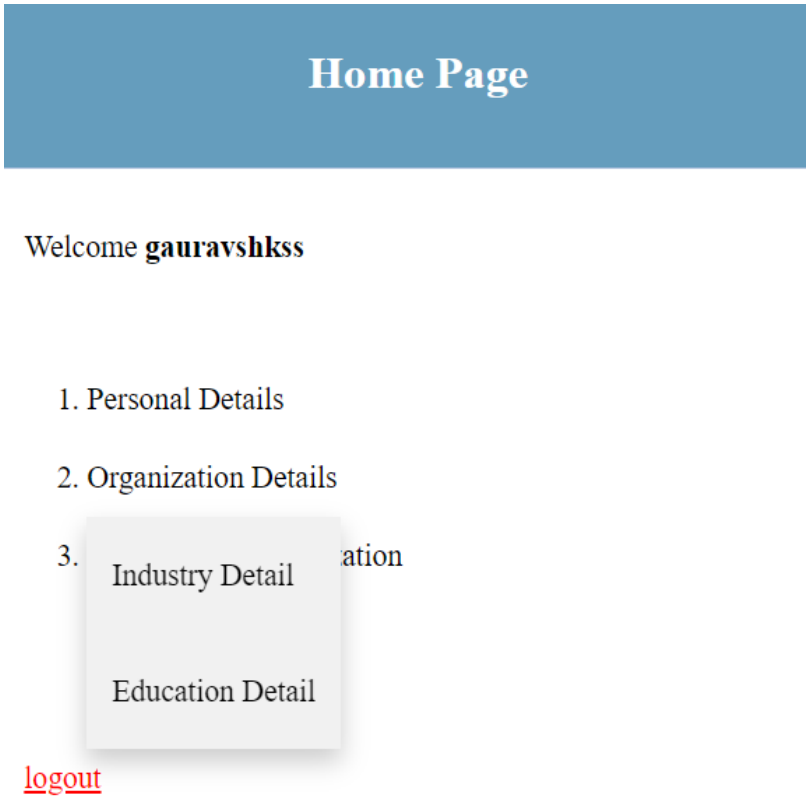
```php
    mysqli_query($db, "UPDATE info SET name='$name', address='$address', profession = '$profession',dob = $dob, phone = '$phone'
        WHERE id=$id");
    $_SESSION['message'] = "Updated successfully!";
    header('location: personalhome.php');
}

    if (isset($_GET['del'])) {
    $id = $_GET['del'];
    mysqli_query($db, "DELETE FROM info WHERE id=$id");
    $_SESSION['message'] = "Address deleted!";
    header('location: personalhome.php');
}
```

4.

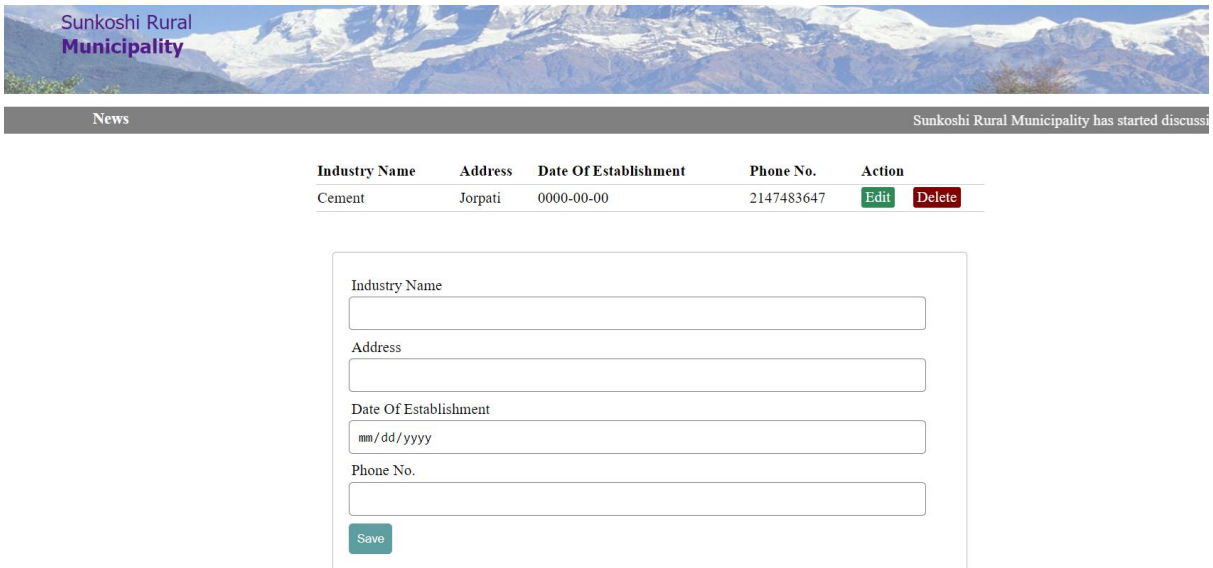
gallery.php | aboutus.php | crudindex.php

```css
body {
    font-size: 19px;
}
table{
    width: 50%;
    margin: 30px auto;
    border-collapse: collapse;
    text-align: left;
}
tr {
    border-bottom: 1px solid #cbcbcb;
}
th, td{
    border: none;
    height: 30px;
    padding: 2px;
}
tr:hover {
    background: #F5F5F5;
}

form {
    width: 45%;
    margin: 50px auto;
    text-align: left;
    padding: 20px;
    border: 1px solid #bbbbbb;
    border-radius: 5px;
}

.input-group {
    margin: 10px 0px 10px 0px;
}
.input-group label {
    display: block;
    text-align: left;
    margin: 3px;
}
}
.input-group input {
    height: 30px;
    width: 93%;
    padding: 5px 10px;
    font-size: 16px;
    border-radius: 5px;
    border: 1px solid gray;
}
.btn {
    padding: 10px;
    font-size: 15px;
    color: white;
    background: #5F9EA0;
    border: none;
    border-radius: 5px;
}
.edit_btn {
    text-decoration: none;
    padding: 2px 5px;
    background: #2E8B57;
    color: white;
    border-radius: 3px;
}

.del_btn {
    text-decoration: none;
    padding: 2px 5px;
    color: white;
    border-radius: 3px;
    background: #800000;
}
```

```
}
.msg {
    margin: 30px auto;
    padding: 10px;
    border-radius: 5px;
    color: #3c763d;
    background: #dff0d8;
    border: 1px solid #3c763d;
    width: 50%;
    text-align: center;
}
.bgimage #bg{

    margin: -8px;
    opacity: 0.6;
}
#bg {

}
.bgimage #flag{
    width: 30%;
}
.bgimage p{
    position: absolute;

     top: -20px;
}
#bg {
    max-width: 101.5%;
    min-height: 150px;

}
#topnav{
    width: 100%;
}
```



Industry Detail:



Code:

1.

```php
<?php  include('industryphp.php'); ?>
    <?php
    if (isset($_GET['edit'])) {
        $id = $_GET['edit'];
        $update = true;
        $record = mysqli_query($db, "SELECT * FROM industry WHERE id=$id");

        if (@count($record) == 1 ) {
            $n = mysqli_fetch_array($record);
            $name = $n['name'];
            $address = $n['address'];

            $date = $n['date'];
            $phone = $n['phone'];
        }
    }
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="industrystyle.css">
    <link rel="icon" type="image/jpg" href="icon.ico">
    <title>Organization Details</title>

    <link rel="stylesheet" type="image/jpg" href="flag.icon">
</head>
<body>
    <?php include('header.php'); ?>
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
        ?>
    </div>
<?php endif ?>
    <?php $results = mysqli_query($db, "SELECT * FROM industry"); ?>

    <table>
    <thead>
        <tr>
            <th>Industry Name</th>

            <th>Address</th>

            <th>Date Of Establishment</th>
            <th>Phone No.</th>
```

```
            <th colspan="2">Action</th>
        </tr>
    </thead>

    <?php while ($row = mysqli_fetch_array($results)) { ?>
        <tr>
            <td><?php echo $row['name']; ?></td>
            <td><?php echo $row['address']; ?></td>
            <td><?php echo $row['date']; ?></td>
            <td><?php echo $row['phone']; ?></td>

            <td>
                <a href="industryhome.php?edit=<?php echo $row['id']; ?>" class="edit_btn" >Edit</a>
            </td>
            <td>
                <a href="industryphp.php?del=<?php echo $row['id']; ?>" class="del_btn">Delete</a>
            </td>
        </tr>
    <?php } ?>
</table>

    <form method="post" action="industryphp.php" >
        <input type="hidden" name="id" value="<?php echo $id; ?>">
        <div class="input-group">
            <label>Industry Name</label>
            <input type="text" name="name" value="<?php echo $name; ?>">
        </div>
        <div class="input-group">
            <label>Address</label>
            <input type="text" name="address" value="<?php echo $address; ?>">
        </div>

        <div class="input-group">
            <label>Date Of Establishment</label>
            <input type="Date" name="date" value="<?php echo $date; ?>">
        </div>
        <div class="input-group">
            <label>Phone No.</label>
            <input type="text" name="phone" value="<?php echo $phone; ?>">

        </div>
```

```
        <div class="input-group">
            <?php if ($update == true): ?>
    <button class="btn" type="submit" name="update" style="background: #556B2F;" >update</button>
<?php else: ?>
    <button class="btn" type="submit" name="save" >Save</button>
<?php endif ?>
        </div>
    </form>


<?php include ('footer.php'); ?>


</body>
</html>
```

2.

```php
<?php  include('industryphp.php'); ?>
    <?php
    if (isset($_GET['edit'])) {
        $id = $_GET['edit'];
        $update = true;
        $record = mysqli_query($db, "SELECT * FROM industry WHERE id=$id");

        if (@count($record) == 1 ) {
            $n = mysqli_fetch_array($record);
            $name = $n['name'];
            $address = $n['address'];
            $phone = $n['phone'];
            $date = $n['date'];
        }
    }
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="industrystyle.css">
    <title>Personal Details</title>

    <link rel="stylesheet" type="image/jpg" href="flag.icon">
</head>
<body>
    <div class="">
        <div class="topnav">
        <div class="bgimage">
            <img id="bg" src="BG.jpg">
            <p><img id="flag" src="nepalflag.gif"></p>

        </div>
    </div>

    </div>
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
        ?>
    </div>
<?php endif ?>
    <?php $results = mysqli_query($db, "SELECT * FROM info"); ?>
```

```php
<table>
<thead>
    <tr>
        <th>Name</th>
        <th>Address</th>
        <th>Phone No.</th>
        <th>Date of establish</th>
        <th colspan="2">Action</th>
    </tr>
</thead>

<?php while ($row = mysqli_fetch_array($results)) { ?>
    <tr>
        <td><?php echo $row['name']; ?></td>
        <td><?php echo $row['address']; ?></td>
        <td><?php echo $row['phone']; ?></td>
        <td><?php echo $row['date']; ?></td>

        <td>
            <a href="industryhome.php?edit=<?php echo $row['id']; ?>" class="edit_btn" >Edit</a>
        </td>
        <td>
            <a href="industryphp.php?del=<?php echo $row['id']; ?>" class="del_btn">Delete</a>
        </td>
    </tr>
<?php } ?>
/table>

<form method="post" action="industryphp.php" >
    <input type="hidden" name="id" value="<?php echo $id; ?>">
    <div class="input-group">
        <label>Name</label>
        <input type="text" name="name" value="<?php echo $name; ?>">
    </div>
```

```
        <div class="input-group">
            <label>Address</label>
            <input type="text" name="address" value="<?php echo $address; ?>">
        </div>
        <div class="input-group">
            <label>Phone No.</label>
            <input type="text" name="phone" value="<?php echo $phone; ?>">

        </div>
        <div class="input-group">
            <label>Date of establishment</label>
            <input type="text" name="date" value="<?php echo $date; ?>">

        </div>

        <div class="input-group">
            <?php if ($update == true): ?>
    <button class="btn" type="submit" name="update" style="background: #556B2F;" >update</button>
<?php else: ?>
    <button class="btn" type="submit" name="save" >Save</button>
<?php endif ?>
        </div>
    </form>




</body>
</html>
```

3.

```
<?php
    session_start();
    $db = mysqli_connect('localhost', 'root', '', 'crud');

    // initialize variables
    $name = "";
    $address = "";
    $id = 1;
    $phone = "";

    $date="";
    $update = false;

    if (isset($_POST['save'])) {
        $name = $_POST['name'];
        $address = $_POST['address'];

        $date = $_POST['date'];
        $phone = $_POST['phone'];

        mysqli_query($db, "INSERT INTO industry (name, address,date,phone) VALUES ('$name', '$address',$date,'$phone')");
        $_SESSION['message'] = "Saved successfully :)";
        header('location: industryhome.php');
    }

        if (isset($_POST['update'])) {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $address = $_POST['address'];
    $profession = $_POST['profession'];
    $date = $_POST['date'];
    $phone = $_POST['phone'];

    mysqli_query($db, "UPDATE industry SET name='$name', address='$address',date = $date, phone = '$phone' WHERE id=$id");
    $_SESSION['message'] = "Updated successfully!";
    header('location: personal.php');
}

    if (isset($_GET['del'])) {
    $id = $_GET['del'];
    mysqli_query($db, "DELETE FROM industry WHERE id=$id");
    $_SESSION['message'] = "Address deleted!";
    header('location: personal.php');
}
```

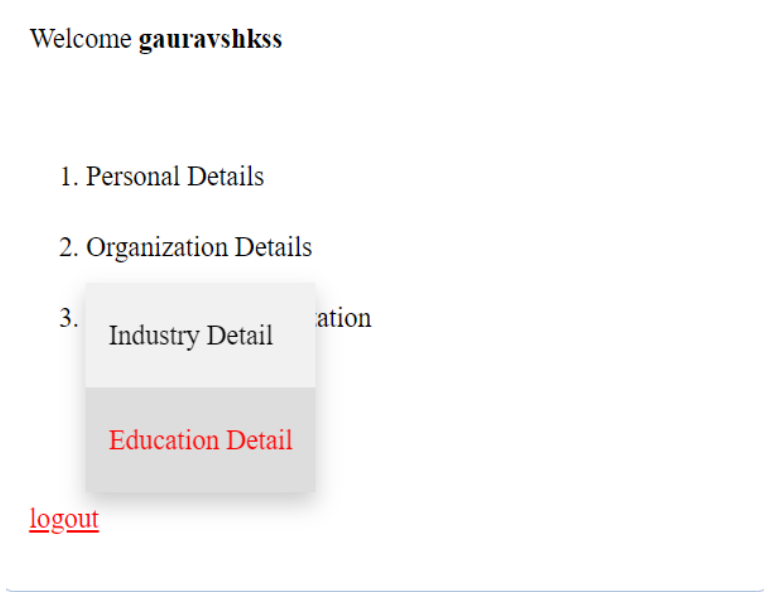
4.

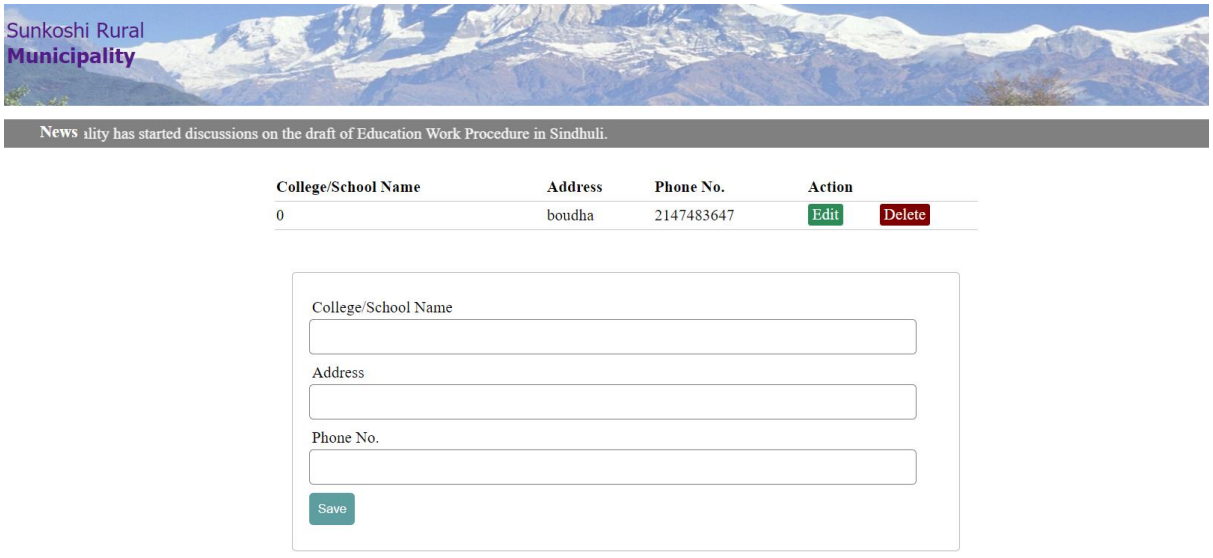
```css
body {
    font-size: 19px;
}
table{
    width: 50%;
    margin: 30px auto;
    border-collapse: collapse;
    text-align: left;
}
tr {
    border-bottom: 1px solid #cbcbcb;
}
th, td{
    border: none;
    height: 30px;
    padding: 2px;
}
tr:hover {
    background: #F5F5F5;
}

form {
    width: 45%;
    margin: 50px auto;
    text-align: left;
    padding: 20px;
    border: 1px solid #bbbbbb;
    border-radius: 5px;
}

.input-group {
    margin: 10px 0px 10px 0px;
}
.input-group label {
    display: block;
    text-align: left;
    margin: 3px;
}
.input-group input {
    height: 30px;
    width: 93%;
    padding: 5px 10px;
    font-size: 16px;
    border-radius: 5px;
    border: 1px solid gray;
}
```

```css
.btn {
    padding: 10px;
    font-size: 15px;
    color: white;
    background: #5F9EA0;
    border: none;
    border-radius: 5px;
}
.edit_btn {
    text-decoration: none;
    padding: 2px 5px;
    background: #2E8B57;
    color: white;
    border-radius: 3px;
}

.del_btn {
    text-decoration: none;
    padding: 2px 5px;
    color: white;
    border-radius: 3px;
    background: #800000;
}
.msg {
    margin: 30px auto;
    padding: 10px;
    border-radius: 5px;
    color: #3c763d;
    background: #dff0d8;
    border: 1px solid #3c763d;
    width: 50%;
    text-align: center;
}
```

```css
.bgimage #bg{

    margin: -8px;
    opacity: 0.6;
}
#bg {

}
.bgimage #flag{
    width: 30%;
}
.bgimage p{
    position: absolute;

     top: -20px;
}
#bg {
    max-width: 101.5%;
    min-height: 150px;

}
#topnav{
    width: 100%;
}
```

For education details:





1.

```php
<?php  include('educationphp.php'); ?>
    <?php
    if (isset($_GET['edit'])) {
        $id = $_GET['edit'];
        $update = true;
        $record = mysqli_query($db, "SELECT * FROM education WHERE id=$id");

        if (@count($record) == 1 ) {
            $n = mysqli_fetch_array($record);
            $name = $n['name'];
            $address = $n['address'];


            $phone = $n['phone'];
        }
    }
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="educationstyle.css">
    <link rel="icon" type="image/jpg" href="icon.ico">
    <title>Organization Details</title>

    <link rel="stylesheet" type="image/jpg" href="flag.icon">
</head>
<body>
    <?php include('header.php'); ?>
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
        ?>
    </div>
<?php endif ?>
    <?php $results = mysqli_query($db, "SELECT * FROM education"); ?>

    <table>
    <thead>
        <tr>
            <th>College/School Name</th>
            <th>Address</th>
            <th>Phone No.</th>
            <th colspan="2">Action</th>
        </tr>
```

```
        </tr>
    </thead>

    <?php while ($row = mysqli_fetch_array($results)) { ?>
        <tr>
            <td><?php echo $row['name']; ?></td>
            <td><?php echo $row['address']; ?></td>

            <td><?php echo $row['phone']; ?></td>

            <td>
                <a href="educationhome.php?edit=<?php echo $row['id']; ?>" class="edit_btn" >Edit</a>
            </td>
            <td>
                <a href="educationphp.php?del=<?php echo $row['id']; ?>" class="del_btn">Delete</a>
            </td>
        </tr>
    <?php } ?>
</table>

    <form method="post" action="educationphp.php" >
        <input type="hidden" name="id" value="<?php echo $id; ?>">
        <div class="input-group">
            <label>College/School Name</label>
            <input type="text" name="name" value="<?php echo $name; ?>">
        </div>
        <div class="input-group">
            <label>Address</label>
            <input type="text" name="address" value="<?php echo $address; ?>">
        </div>


        <div class="input-group">
            <label>Phone No.</label>
            <input type="text" name="phone" value="<?php echo $phone; ?>">

        </div>
        <div class="input-group">
            <?php if ($update == true): ?>
    <button class="btn" type="submit" name="update" style="background: #556B2F;" >update</button>
<?php else: ?>
    <button class="btn" type="submit" name="save" >Save</button>
<?php endif ?>
        </div>
    </form>
<?php include ('footer.php'); ?>
</body>
</html>
```

2.

```php
<?php include('educationphp.php'); ?>
    <?php
    if (isset($_GET['edit'])) {
        $id = $_GET['edit'];
        $update = true;
        $record = mysqli_query($db, "SELECT * FROM education WHERE id=$id");

        if (@count($record) == 1 ) {
            $n = mysqli_fetch_array($record);
            $name = $n['name'];
            $address = $n['address'];
            $phone = $n['phone'];

        }
    }
?>

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="industrystyle.css">
    <title>Education Details</title>

    <link rel="stylesheet" type="image/jpg" href="flag.icon">
</head>
<body>
    <div class="">
        <div class="topnav">
        <div class="bgimage">
            <img id="bg" src="BG.jpg">
            <p><img id="flag" src="nepalflag.gif"></p>

        </div>
    </div>

    </div>
    <?php if (isset($_SESSION['message'])): ?>
    <div class="msg">
        <?php
            echo $_SESSION['message'];
            unset($_SESSION['message']);
        ?>
    </div>
<?php endif ?>
    <?php $results = mysqli_query($db, "SELECT * FROM info"); ?>
```

```
    <table>
    <thead>
        <tr>
            <th>Name</th>
            <th>Address</th>
            <th>Phone No.</th>
            <th colspan="2">Action</th>
        </tr>
    </thead>
    <?php while ($row = mysqli_fetch_array($results)) { ?>
        <tr>
            <td><?php echo $row['name']; ?></td>
            <td><?php echo $row['address']; ?></td>
            <td><?php echo $row['phone']; ?></td>
            <td>
                <a href="educationhome.php?edit=<?php echo $row['id']; ?>" class="edit_btn" >Edit</a>
            </td>
            <td>
                <a href="educationphp.php?del=<?php echo $row['id']; ?>" class="del_btn">Delete</a>
            </td>
        </tr>
    <?php } ?>
</table>
    <form method="post" action="educationphp.php" >
        <input type="hidden" name="id" value="<?php echo $id; ?>">
        <div class="input-group">
            <label>Name</label>
            <input type="text" name="name" value="<?php echo $name; ?>">
        </div>
        <div class="input-group">
            <label>Address</label>
            <input type="text" name="address" value="<?php echo $address; ?>">
        </div>
        <div class="input-group">
            <label>Phone No.</label>
            <input type="text" name="phone" value="<?php echo $phone; ?>">
        </div>
        <div class="input-group">
            <?php if ($update == true): ?>
    <button class="btn" type="submit" name="update" style="background: #556B2F;" >update</button>
<?php else: ?>
    <button class="btn" type="submit" name="save" >Save</button>
<?php endif ?>
        </div>
    </form>
</body>
</html>
```

3.

```php
<?php
    session_start();
    $db = mysqli_connect('localhost', 'root', '', 'crud');

    // initialize variables
    $name = "";
    $address = "";
    $id = 1;
    $phone = "";


    $update = false;

    if (isset($_POST['save'])) {
        $name = $_POST['name'];
        $address = $_POST['address'];


        $phone = $_POST['phone'];

        mysqli_query($db, "INSERT INTO education (name, address,phone) VALUES ('$name', '$address','$phone')");
        $_SESSION['message'] = "Saved successfully :)";
        header('location: educationhome.php');
    }

        if (isset($_POST['update'])) {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $address = $_POST['address'];
    $phone = $_POST['phone'];

    mysqli_query($db, "UPDATE education SET name='$name', address='$address', phone = '$phone' WHERE id=$id");
    $_SESSION['message'] = "Updated successfully!";
    header('location: personal.php');
}

    if (isset($_GET['del'])) {
    $id = $_GET['del'];
    mysqli_query($db, "DELETE FROM education WHERE id=$id");
    $_SESSION['message'] = "Address deleted!";
    header('location: personal.php');
}
```

4.

```css
body {
    font-size: 19px;
}
table{
    width: 50%;
    margin: 30px auto;
    border-collapse: collapse;
    text-align: left;
}
tr {
    border-bottom: 1px solid #cbcbcb;
}
th, td{
    border: none;
    height: 30px;
    padding: 2px;
}
tr:hover {
    background: #F5F5F5;
}

form {
    width: 45%;
    margin: 50px auto;
    text-align: left;
    padding: 20px;
    border: 1px solid #bbbbbb;
    border-radius: 5px;
}

.input-group {
    margin: 10px 0px 10px 0px;
}
.input-group label {
    display: block;
    text-align: left;
    margin: 3px;
}
.input-group input {
    height: 30px;
    width: 93%;
    padding: 5px 10px;
    font-size: 16px;
    border-radius: 5px;
    border: 1px solid gray;
}
```

```css
.btn {
    padding: 10px;
    font-size: 15px;
    color: white;
    background: #5F9EA0;
    border: none;
    border-radius: 5px;
}
.edit_btn {
    text-decoration: none;
    padding: 2px 5px;
    background: #2E8B57;
    color: white;
    border-radius: 3px;
}
.del_btn {
    text-decoration: none;
    padding: 2px 5px;
    color: white;
    border-radius: 3px;
    background: #800000;
}
.msg {
    margin: 30px auto;
    padding: 10px;
    border-radius: 5px;
    color: #3c763d;
    background: #dff0d8;
    border: 1px solid #3c763d;
    width: 50%;
    text-align: center;
}
.bgimage #bg{
    margin: -8px;
    opacity: 0.6;
}
.bgimage #flag{
    width: 30%;
}
.bgimage p{
    position: absolute;
     top: -20px;
}
#bg {
    max-width: 101.5%;
    min-height: 150px;
}
```

Error.php:

```php


  <?php if (count($errors) > 0) : ?>
    <div class="error">
      <?php foreach ($errors as $error) : ?>
        <p><?php echo $error ?></p>
      <?php endforeach ?>
    </div>
  <?php  endif ?>
```

Footer:

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
    <style type="text/css">


    .logo{
        margin-top: 100px;
        background-color: #05B1F8;

        margin-left: -8px;
        margin-right: -8px;
    }
    .footer{

    }
    .footer .logo{
        text-align: center;
    }
    #medialogo:hover{
    opacity: 0.8;
    transform: scale(1.5);
    transition: 1s;
    }
    .logo p{
        font-size: 17px;
    }
    .contact{
        float: left;
        font-size: 15px;
        font-family: sans-serif;
        color: black;
        font-weight: bolder;
        border-right: 20px;

    }
    .link{
        width: 90%;
        float: center;

    }
    </style>

    </style>
```

```
    </style>
<div class = footer>

        <div class="logo">
            <div class="contact">

                <b><p>Contacts</p></b>
                Toll free number: 1583659375 <br> <br>
                Phone No: +9779860725577

            </div>
            <div class="link">
                <p>Keep in touch</p>
            <a href="https://www.facebook.com/gaurav.shakya1n2" target="_Blank"><img id="medialogo" width="25px;" src="fb.png"></a>
        <a href="https://www.twitter.com/Gauravs45230259" target="_Blank"><img id="medialogo" width="25px;" src="twitter.png"></a>
        <a href="https://www.instagram.com/gauravshk/" target="_Blank"><img id="medialogo" width="27px;" src="insta.png"></a>
        <a href="https://www.linkedin.com/in/gaurav-shakya-82373a184/" target="_Blank"><img id="medialogo" width="25px;" src="linkedin.png"></a>
            </div>

            <br><br><br><br><br><br>
        </div>


    </div>
</body>
</html>
```

Conclusion

Finally, I have developed a complete website following the overall process of software development from the requirements specification to development and testing. This project is for a Sunkoshi Rural Municipality which is in Sindhupalchowk district where they wanted an online system of data collection along with the functionalities like view the records, edit and delete as per need. Data collection are categorized as: personal, organizational, industrial etc. Personal details includes name, address, profession, date of birth etc. Organizational are broken down into government, educational, hospitals etc. Record entry form, its view, edit, delete functionalities are different for different categories.

From the requirements above, I have created a fidelity wireframes with full set of client and user requirements. After this, this website contains more than 10 pages and have CRUD functionality. At last, I have created test plan to test my developed website identifying key performance areas and use it to review the overall functionality and performance of the website.

Bibliography

Anon., n.d. [Online]
Available at: schools.com/html/default.asp

Anon., n.d. [Online]
Available at: http://www.site-ninja.com/w3c-validator

Anon., n.d. *cloudflare.* [Online]
Available at: https://www.cloudflare.com/learning/bots/what-is-a-web-crawler/

Anon., n.d. *Searchengineland.* [Online]
Available at: https://searchengineland.com/guide/what-is-seo

Anon., n.d. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/sap_webi/sap_webi_ranking_data.htm

Anon., n.d. *wireframe.* [Online]
Available at: https://wireframe.cc/