

**Final Report**  
on  
**EDUCATIONAL NOTES SHARING WEBSITE**

*Submitted in partial fulfillment of the  
requirement for the award of the degree of*



**Under The Supervision of  
Dr. Aanjey Mani Tripathi  
Assistant Professor**

Submitted By

Abhishek Singh (19SCSE1010290)  
Nikhil Sharma (19SCSE1010128)

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING / DEPARTMENT OF  
COMPUTERAPPLICATION  
GALGOTIAS UNIVERSITY, GREATER NOIDA  
INDIA  
DEC, 2021**



# **SCHOOL OF COMPUTING SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA**

## **CANDIDATE'S DECLARATION**

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled “**EDUCATIONAL NOTES SHARING WEBSITE**” in partial fulfillment of the requirements for the award of the **BACHELOR IN COMPUTER SCIENCE** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of **JUNE - 2021** to **DEC - 2021**, under the supervision of **DR. AANJEY MANI TRIPATHI** , **Assistant Professor** , Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

**Abhishek Singh (19SCSE1010290)**

**Nikhil Sharma (19SCSE1010128)**

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Aanjey Mani Tripathi

Assistant Professor

**CERTIFICATE**

The Final Thesis/Project/ Dissertation Viva-Voce examination of **Abhishek Singh(19SCSE1010290)**  
**Nikhil Sharma (19SCSE1010128)** has been held on \_\_\_\_\_ and his/her work is recommended  
for the award of **Bachelor in computer science and engineering**.

**Signature of Examiner(s)**

**Signature of Supervisor(s)**

**Signature of Project Coordinator**

**Signature of Dean**

Date: **DEC , 2021**

Place: Greater Noida

## TABLE OF CONTENTS

<b>SNO.</b>	<b>TITLE</b>	<b>Page No.</b>
<b>01</b>	<b>Abstract</b>	<b>02</b>
<b>02</b>	<b>Introduction</b>	<b>03</b>
<b>03</b>	<b>Literature Survey</b>	<b>04</b>
<b>04</b>	<b>Problem Statement</b>	<b>05</b>
<b>05</b>	<b>Proposed Approach</b>	<b>06</b>
<b>06</b>	<b>HTML</b>	<b>07</b>
<b>07</b>	<b>CSS</b>	<b>19</b>
<b>08</b>	<b>SQL</b>	<b>24</b>
<b>09</b>	<b>Django</b>	<b>28</b>
<b>10</b>	<b>Implementation</b>	<b>39</b>
<b>11</b>	<b>Result</b>	<b>47</b>
<b>12</b>	<b>Testing</b>	<b>51</b>
<b>13</b>	<b>Refrences</b>	<b>53</b>

## **Abstract**

Most of the students due to lack of communication from classmates or Senior's face problem of not getting class notes, if were absent from class or any new subject has been added to the curriculum this problem has increased because of strict lockdown, colleges been closed for physical learning increasing the gap between students.

Taking this problem into consideration the idea of this website was developed. It is designed to simplify and ease the process of study. In this website a user can also upload Notes if they are not available or either a new subject is added to curriculum to keep the website updated. The notes that are uploaded by the user will only be displayed on the website with the approval of the Administrator to maintain the integrity of website.

This website has been developed by using HTML, CSS, J Query and Bootstrap for Frontend development and for backend we have used python, Django and for database we have used My Sql .

This website will simplify the process of study for the student and will be very helpful for them to get notes and syllabus for course subject and also with the subject that are newly added to their course.

Because of use of this website student will be able to get Study material easily. And quiz tab can also be added to website which can contain objective question by which students can better analyze of their knowledge.

## **Chapter I. INTRODUCTION**

Technology has played important role in improving and expanding education. It has Simplified and eased the process of study. The availability of education resource has helped students a lot in their educational life.

There are many websites that share educational resources like digital notes or videolecture but none of them provide class notes and syllabus and also newly added subject are also not added to these websites in timely manner. That can create problem for the student. And teachers also have problem to share their notes to all the students.

The idea of this website is to provide all class notes of different courses with an option for the user to upload new notes for newly added subject or Existing subject to keep the website because of this feature students can upload notes and teacher as well can upload notes to website to reach to maximum student. The documents uploaded by users (Students/Teachers) will only appear on the website when there Approved by the Administrator for their Authenticity.

This website is being developed by using HTML, CSS, J Query and Bootstrap in Frontend. And for backend development we have used Python, Django and for database development we have used My SQL.

## Chapter II. LITERATURE SURVEY

Different website has been developed on the idea of educational resource sharing some of them are listed below: -

**1. Aglasem:**

This platform provides previous year question paper for different exams.

**2. Tutorials Point:**

The website provides tutorials for different topics related to Computer science and technology

**3. JavaTpoint**

This website provides tutorials on different programming language and Concepts of computer Science.

**4. W3schools**

This website provides tutorials of different programming language.

**4. The Physics Classroom:**

For PDF notes and tutorials related to the various fields of Physics.

**5. Aglasem:**

Online Portal that provides previous year question papers related to different competitive exams and some universities semester papers.

**6. A Django Based Educational Resource Sharing Website Shreic:**

This Platform provide the features to download and view Notes.

**7. Vioric-Torri, C. & Alexandrache, C. (2012):**

The study reflects how educational technology influences the learning styles of students and how to form and develop the competences of learning in the new generations.

After analyzing all the previous work, it has been observed that all these websites possess different function. There is no such website that has to the point materials for engineering Subject and allow users to upload notes if not available on platform for any newly added Subject to the curriculum to contribute to the database.

## Chapter III. Problem Statement

In the time of covid and in general the student need to access the notes of any subject with an ease there are many website that provide the materials of subjects online. But, none of them provide class notes or question papers related to courses taught in a particular college or university; and most of the students due to lack of communication from batch mates or seniors, often face the problem of not getting class notes, if were absent or previous year question papers from seniors.

## Chapter IV. PROPOSED APPROACH

### A. *Software Development Life Cycle*

Software Development is the development of software for distinct purposes. For software development, there is a specific programming language like Java, Python, C/C++, etc. The entire process of software development isn't as simple as its definition, it's a complicated process. Accordingly, it requires an efficient approach from the developer in the form of the Software Development Life Cycle (SDLC).

Proper planning and execution are the key components of a successful software development process. The entire software development process includes 6 stages. Software Development Life Cycle (SDLC) is the common term to summarize these 6 stages.

SDLC specifies the task(s) to be performed at various stages by a software engineer/developer. It ensures that the end product is able to meet the customer's expectations and fits in the overall budget. Hence, it's vital for a software developer to have prior knowledge of this software development process.

These 6 stages are discussed below:

- **Stage-1: Planning and Requirement Analysis:**

Planning is the crucial step in everything and so as in software development. In this same stage, requirement analysis is also performed by the developers of the organization. This is attained from the inputs from the customers, sales department/market surveys.

The information from this analysis forms the building block of a basic project. The quality proof of the project is a result of planning. Thus, in this stage, the basic project is designed with all the available information.

- **Stage-2: Defining Requirements:**

In this stage, all the requirements for the target software are specified. These requirements get approval from the customers, market analysts, and stakeholders.

This is fulfilled by utilizing SRS (Software Requirement Specification). This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

- **Stage-3: Designing Architecture:**

SRS is a reference for software designers to come out with the best architecture for the software. Hence, with the requirements defined in SRS, multiple designs for the product architecture are present in the Design Document Specification (DDS).

This DDS is assessed by market analysts and stakeholders. After evaluating all the possible factors, the

most practical and logical design is chosen for the development.

- **Stage-4: Developing Product:**

At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association. Conventional programming tools like compilers, interpreters, debuggers, etc. are also put into use at this stage. Some popular languages like C/C++, Python, Java, etc. are put into use as per the software regulations.

- **Stage-5: Product Testing and Integration:**

After the development of the product, testing of the software is necessary to ensure its smooth execution. Although, minimal testing is conducted at every stage of SDLC.

Therefore, at this stage, all the probable flaws are tracked, fixed, and retested. This ensures that the product confronts the quality requirements of SRS.

- **Documentation, Training and Support:**

Software documentation is an essential part of the software development life cycle. A well-written document acts as a tool and means to information repository necessary to know about software processes, functions and maintenance. Documentation also provides information about how to use the product. Thoroughly-written documentation should involve the required documentation. Software architecture documentation, technical documentation and user documentation. Training in an attempt to improve the current or future employee performance by increasing an employee's ability to work through learning, usually by changing his attitude and developing his skills and understanding.

- **Stage 6: Deployment and Maintenance of Product:**

After detailed testing, the conclusive product is released in phases as per the organization's strategy. Then it is tested in a real industrial environment. Because it is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough. Therefore, along with the deployment, the product's supervision.

## ***B. SDLC Model***

A software life cycle model is a descriptive representation of the software development cycle. The software development model helps the developer to select a strategy to develop the software. A software development model has its own set of tools, methods and procedures, which are expressed clearly and defines the software development life cycle. This project has been developed using the Iterative model (Jalote, 2003).

In this life cycle model, a Project Control List (PCL) on the basis of current known requirements is developed. A PCL is a list containing the series of tasks/functionalities that are to be present in the given system. If at a certain phase of development, we come across any new requirement, we add it to our Project Control List.

For developing the website, a task is chosen from the given PCL and Planning, Analysis, Designing, Testing and Evaluation is performed as shown in Figure 1. When the specific functionality is added we remove it from the



Project Control List. In a similar way, one task at a time from PCL is chosen, implemented and then removed from PCL. This process iterates until the desired requirements of the product are not met.

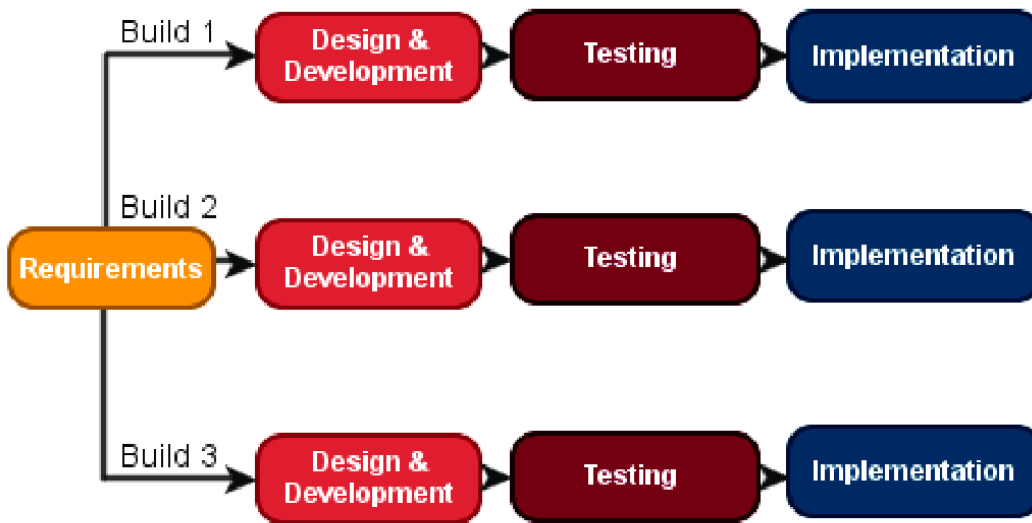


Fig. 1. Iterative Model

After each iteration, the management team can do work on risk management and prepare for the next iteration. Because a cycle includes a small portion of the whole software process, it is easier to manage the development process.

In the Iterative model, the newer iterations are incrementally improved versions of previous iterations. Moreover, in the event that a new iteration fundamentally breaks a system in a catastrophic manner, a previous iteration can quickly and easily be implemented or “rolled back,” with minimal losses, which is a boon for post-release maintenance.

In the Iterative Model, the initial run-through of all stages may take some time, but each subsequent iteration will be faster and faster, allowing the life cycle of each new iteration to be trimmed down to a matter of days or even hours in some cases.

### ***Feasibility Study***

Feasibility analysis determines the viability of an idea, like making sure the project is technically, economically and legally.

At the time of development of this project the feasibility study was done as following: -

## 1)Project Requirements

These are the following objective that were proposed at the time of development :-

- Administrator Login
- Different Categories for different branch
- Different sub category under branch which has different option of semesters.
- An uploading function for the user to upload the Notes.
- Administrator can upload or delete notes.
- Administrator can Approve the notes uploaded by the user.

This requirement list was also used as the Project Control List during development.

Certain goals regarding the efficiency of the project to be developed were also proposed, which are as follows:

- **Planned Approach:** The working of the website is well planned and organized. The data will be stored properly in data stores, which will help in the retrieval of information as well as its storage.
- **Accuracy:** The level of accuracy in the proposed system will be higher. All operations would be done correctly and it ensures that whatever information is retrieved or stored is accurate.
- **Reliability:** The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.
- **No Redundancy:** In the proposed system utmost care would be taken so that no information is repeated anywhere, in storage or otherwise. This would assure the economic use of storage space and consistency in the data stored.
- **Immediate retrieval of information:** The main objective of the proposed system is to provide quick and efficient retrieval of information regarding users, orders, products etc.
- **Easy to Operate:** The system should be user-friendly and should be such that it can be developed within a short period of time and fit in the limited budget of the organization.

## ***E-R Diagram***

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

The project consists of many relational models. Every model consists of different attributes. The Primary Key is shown with an underline under the name. Each Relation is characterized by Structural Constraint where the first number denotes Participation Constraint (1 for Total Participation and 0 for Partial Participation) and the second number denotes the Cardinality Ratio (1:1 or 1: N or M: N).

### *1) Participation Constraint:*

Total Participation: If each entity of an Entity Type has a relationship instance in Relationship Set then the participation is total.

Partial Participation: If few entities of an Entity Type have a relationship instance in Relationship Set then the participation is partial.

2) *Cardinality Ratio:*

1:1: Only one entity of an Entity Set can be related to anyone entity of the other Entity Set.

M: N: Many entities of an Entity Set can be related to many entities of the other Entity Set.

1: N: One entity of an Entity Set can be related to any number of entities of the other Entity Set.

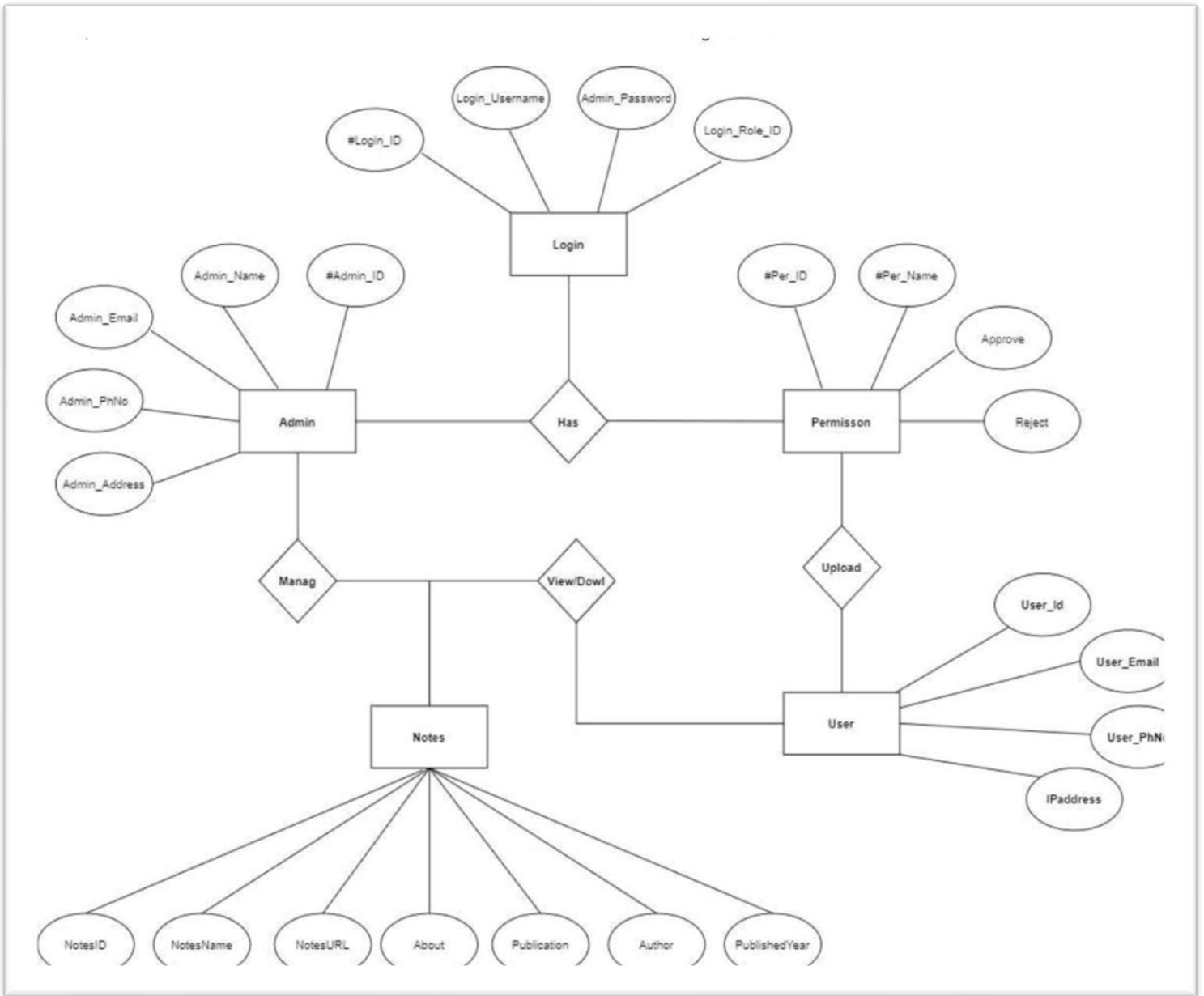


Fig. 2. ER Diagram

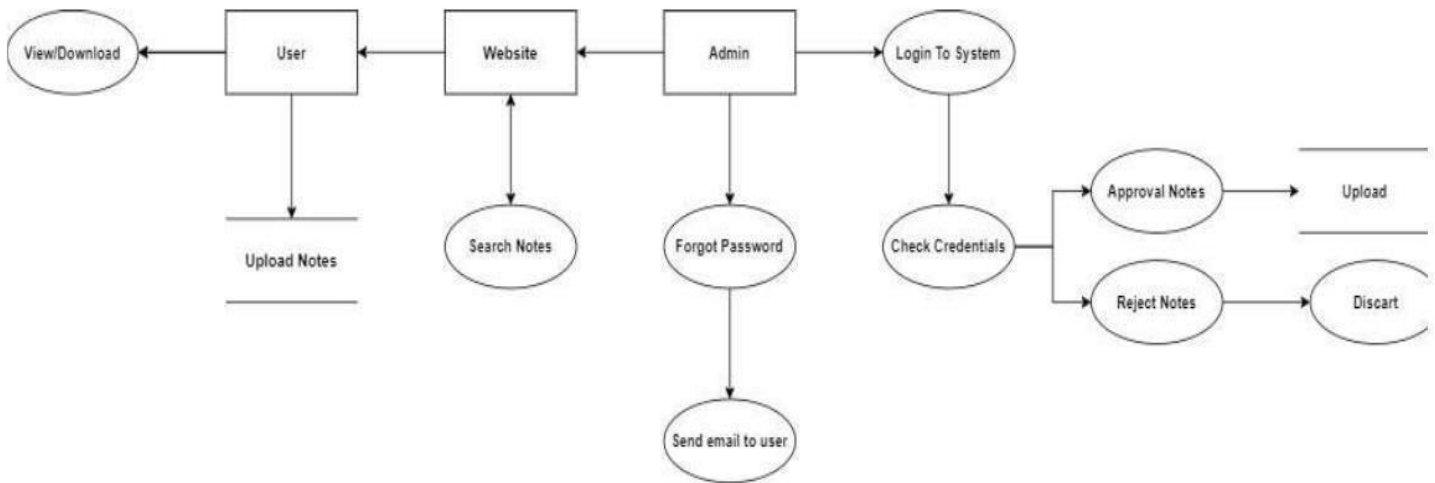
### D. Data Flow

Data flow diagrams were popularized in the late 1970s, arising from the book *Structured Design*, by computing pioneers Ed Yourdon and Larry Constantine. They based it on the “data flow graph” computation models by David Martin and Gerald Estrin. The structured design concept took off in the software engineering field, and the DFD method took off with it. It became more popular in business circles, as it was applied to business analysis, than in academic circles.

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

10/10/21, 1:34 AM

Data Flow Diagram



## HTML

HTML, or HyperText Markup Language, allows web users can create and structure sections, paragraphs, and links using elements, tags, and attributes. However, it's worth noting that HTML is not considered a programming language as it can't create dynamic functionality.

HTML has a lot of use cases, namely:

- Web development. Developers use HTML code to design how a browser displays web page elements, such as text, hyperlinks, and media files.
- Internet navigation. Users can easily navigate and insert links between related pages and websites as HTML is heavily used to embed hyperlinks.
- Web documentation. HTML makes it possible to organize and format documents, similarly to Microsoft Word.

It's also worth noting that HTML is now considered an official web standard. The World Wide Web Consortium (W3C) maintains and develops HTML specifications, along with providing regular updates.

The average website includes several different HTML pages. For instance, a home page, an about page, and a contact page would all have separate HTML files.

HTML documents are files that end with a .html or .htm extension. A web browser reads the HTML file and renders its content so that internet users can view it.

All HTML pages have a series of HTML elements, consisting of a set of tags and attributes. HTML elements are the building blocks of a web page. A tag tells the web browser where an element begins and ends, whereas an attribute describes the characteristics of an element.

The three main parts of an element are:

- Opening tag – used to state where an element starts to take effect. The tag is wrapped with opening and closing angle brackets. For example, use the start tag <p> to create a paragraph.
- Content – this is the output that other users see.

- Closing tag – the same as the opening tag, but with a forward slash before the element name. For example, `</p>` to end a paragraph.

The combination of these three parts will create an HTML element:

```
<p>This is how you add a paragraph in HTML.</p>
```

Another critical part of an HTML element is its attribute, which has two sections – a name and attribute value. The name identifies the additional information that a user wants to add, while the attribute value gives further specifications.

For example, a style element adding the color purple and the font-family verdana will look like this:

```
<p style="color:purple;font-family:verdana">This is how you add a paragraph in HTML.</p>
```

Another attribute, the HTML class, is most important for development and programming. The class attribute adds style information that can work on different elements with the same class value.

For example, we will use the same style for a heading `<h1>` and a paragraph `<p>`. The style includes background color, text color, border, margin, and padding, under the class `.important`. To achieve the same style between `<h1>` and `<p>`, add `class="important"` after each start tag:

```
<html>
<head>
<style>
.important {
background-color: blue;
color: white;
border: 2px solid black;
margin: 2px;
padding: 2px;
}
</style>
</head>
<body>
<h1 class="important">This is a heading</h1>
<p class="important">This is a paragraph.</p>
</body>
```

</html>

Most elements have an opening and a closing tag, but some elements do not need closing tags to work, such as empty elements. These elements do not use an end tag because they do not have content:

```

```

This image tag has two attributes – an src attribute, the image path, and an alt attribute, the descriptive text. However, it does not have content nor an end tag.

Lastly, every HTML document must start with a <!DOCTYPE> declaration to inform the web browser about the document type. With HTML5, the doctype HTML public declaration will be:

```
<!DOCTYPE html>
```

## **Most Used HTML Tags and HTML Elements**

Currently, there are 142 HTML tags available that allow for the creation of various elements. Even though modern browsers no longer support some of these tags, learning all the different elements available is still beneficial.

This section will discuss the most-used HTML tags and two main elements – block-level elements and inline elements.

### **Block-Level Elements**

A block-level element takes up the entire width of a page. It always starts a new line in the document. For example, a heading element will be in a separate line from a paragraph element.

Every HTML page uses these three tags:

- <html> tag is the root element that defines the whole HTML document.
- <head> tag holds meta information such as the page's title and charset.
- <body> tag encloses all the content that appears on the page.

<html>



```
<head>
<!-- META INFORMATION -->
</head>
<body>
<!-- PAGE CONTENT -->
</body>
</html>
```

Other popular block-level tags include:

- Heading tags – these range from `<h1>` to `<h6>`, where heading h1 is largest in size, getting smaller as they move up to h6.
- Paragraph tags – are all enclosed by using the `<p>` tag.
- List tags – have different variations. Use the `<ol>` tag for an ordered list, and use `<ul>` for an unordered list. Then, enclose individual list items using the `<li>` tag.

## Inline Elements

An inline element formats the inner content of block-level elements, such as adding links and emphasized strings. Inline elements are most commonly used to format text without breaking the flow of the content. For example, a `<strong>` tag would render an element in bold, whereas the `<em>` tag would show it in italics. Hyperlinks are also inline elements that use an `<a>` tag and an `href` attribute to indicate the link's destination: `<a href="https://example.com/">Click me!</a>`

## HTML Evolution – What Differs Between HTML and HTML5?

The first version of HTML consisted of 18 tags. Since then, each new version came with new tags and attributes added to the markup. The most significant upgrade of the language so far was the introduction of HTML5 in 2014.

The main difference between HTML and HTML5 is that HTML5 supports new kinds of form controls. HTML5 also introduced several semantic tags that clearly describe the content, such as <article>, <header>, and <footer>.

## **Pros and Cons of HTML**

Just like any other computer language, HTML has its strengths and limitations. Here are the pros and cons of HTML:

Pros:

- Beginner-friendly. HTML has a clean and consistent markup, as well as a shallow learning curve.
- Support. The language is widely used, with a lot of resources and a large community behind it.
- Accessible. It is open-source and completely free. HTML runs natively in all web browsers.
- Flexible. HTML is easily integrable with backend languages such as PHP and Node.js.

Cons:

- Static. The language is primarily used for static web pages. For dynamic functionality, you may need to use JavaScript or a back-end language such as PHP.
- Separate HTML page. Users have to create individual web pages for HTML, even if the elements are the same.
- Browser compatibility. Some browsers adopt new features slowly. Sometimes older browsers don't always render newer tags.

## How Are HTML, CSS, and Javascript Related

HTML is used to add text elements and create the structure of content. However, it is not enough to build a professional and fully responsive website. So, HTML needs the help of Cascading Style Sheets (CSS) and JavaScript to create the vast majority of website content.

CSS is responsible for stylings such as background, colors, layouts, spacing, and animations. On the other hand, JavaScript adds dynamic functionality such as sliders, pop-ups, and photo galleries. These three languages are the fundamentals of front-end development.

## Understanding HTML and Improving Your HTML Knowledge

Learning about HTML is a great first step for those interested in web development.

There are plenty of courses available online to learn to code, but we have listed three of the best tutorial databases for HTML:

- W3Schools – has resources, examples, and exercises to help learn basic HTML for free. There is also a self-paced HTML tutorial that costs \$95 and provides an official certificate.
- Codecademy – offers introductory courses for free with interactive tutorials. Codecademy uses a split-screen that will automatically show the result of your coding on an HTML file. There is exclusive content available for \$19.99/month.
- Coursera – offers various courses that provide in-depth explanations with real-life examples. The subscription price is \$49/month, and there is a 7-day free trial to start.
- HTML is the primary markup language found on the internet. Every HTML page has a series of elements that create the content structure of a web page or application.

- HTML is a beginner-friendly language with plenty of support and is mainly used for static website pages. HTML works best together with CSS for the styling and JavaScript for the functionality. You can check out how to link CSS and HTML on our blog.
- We have also shown you some of the top courses available online that will either help to improve your knowledge of HTML or provide a basic understanding of it.

## CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

---

### What does CSS do

- You can add new looks to your old HTML documents.
- You can completely change the look of your website with only a few changes in CSS code.

---

### Why use CSS

These are the three major benefits of CSS:

#### 1) Solves a big problem

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

#### 2) Saves a lot of time

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

#### 3) Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation.

What is WebFonts?

WebFonts is a technology that enables people to use fonts on demand over the Web without requiring installation in the operating system. W3C has experience in downloadable fonts through HTML, CSS2, and SVG. Until recently, downloadable fonts have not been common on the Web due to the lack of an interoperable font format. The WebFonts effort plans to address that through the creation of an industry-supported, open font format for the Web (called "WOFF").

Examples

The following very simple example of a portion of an HTML document illustrates how to create a link within a paragraph. When rendered on the screen (or by a speech synthesizer), the link text will be “final report”; when somebody activates the link, the browser will retrieve the resource identified by “http://www.example.com/report”:

```
<p class="moreinfo">For more information see the  
<a href="http://www.example.com/report">final report</a>.</p>
```

The class attribute on the paragraph's start tag (“<p>”) can be used, among other things, to add style. For instance, to italicize the text of all paragraphs with a class of “moreinfo,” one could write, in CSS:

```
p.moreinfo { font-style: italic }
```

By placing that rule in a separate file, the style may be shared by any number of HTML documents.

## BOOTSTRAP

### 1. Its responsive grid

No more spending hours coding your own grid—Bootstrap comes with its own grid system predefined.

Now, you can get straight to filling your containers with content.

Defining custom breakpoints for each column is a snap using their extra small, small, medium, large, and extra large breaks. You can also simply stick to the default as it might already meet the needs of your site.

### 2. Its responsive images

Bootstrap comes with its own code for automatically resizing images based on the current screen size. Just add the .img-responsive class to your images, and the predefined CSS rules take care of the rest.

Let Bootstrap resize your images for you!

It can even change the shape of your images with the addition of classes like `img-circle` and `img-rounded`, and that's without going back and forth between the code and your design software.

### 3. Its components

Bootstrap comes with a whole barrelful of components you can easily tack onto your web page, including:

- Navigation bars
- Dropdowns
- Progress bars
- Thumbnails

Not only is it a breeze to add eye-catching design elements to your webpage, you'll also be able to rest assured knowing that every one of them will look great no matter the screen size or device used to view them. That's a lot of ready-made functionality right at your fingertips.

For a more complete list of addable features, check out the component documentation.

### 4. Its JavaScript

Still not enough functions? Try JQuery!

Bootstrap also allows developers to take advantage of over a dozen custom JQuery plugins. This library gives you even more room to play with interactivity, offering up easy solutions for modal popups, transitions, image carousels, and—one of my personal favorites—a plugin called `scrollspy`, which automatically updates your navigation bar as you scroll through a page.

### 5. Its documentation

Simply put, Bootstrap's documentation is some of the best we've ever seen. Every piece of code is described and explained in explicit detail on their website.

Explanations also include code samples for basic implementation, simplifying the process for even the most beginner of beginners. All you need to do is choose a component, copy and paste the code into your page, and tweak from there.

## 6. Its customizability

One of the main critiques when it comes to frameworks such as Bootstrap is their size—the weight they throw around can really slow down your application upon first load. The current version of Bootstrap’s CSS file, for example, is a whopping 119 KB. While this may not seem especially large compared to image and video files, for a CSS file, that’s enormous!

What it allows you to do to combat this, however, is customize which functionality you want to include in your download. By simply going to their [Customize and Download](#) page, you can check off the features you won’t need for your application, trimming the weight off your file and saving your users the additional load time.

Customization is key!

## 7. Its community

As with so many open-source projects, Bootstrap has a large community of designers and developers behind it. Being hosted on GitHub makes it easy for developers to modify and contribute to Bootstrap’s codebase. It also makes it easy for people to collaborate, lend their advice, and interact with peers and fellow users.

Bootstrap has an active Twitter page, a Bootstrap blog, and even a dedicated Slack room. And that doesn’t even get into the wealth of developers willing to help with technical problems on Stack Overflow, where all questions can be found under the `bootstrap-4` tag.

## 8. Its external templates

As its popularity grew, people started creating templates based on Bootstrap in order to accelerate the web development process even further. There are many websites out there dedicated to sharing and buying custom templates based on Bootstrap.

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.
- It facilitates you to create responsive designs.

## History of Bootstrap

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub.

## Why use Bootstrap

Following are the main advantage of Bootstrap:

- It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
- It facilitates users to develop a responsive website.
- It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

## What is a responsive website

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

## What Bootstrap package contains

**Scaffolding:** Bootstrap provides a basic structure with Grid System, link styles, and background.

**CSS:** Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

**Components:** Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.



JavaScript Plugins: Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.

Customize: Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.

What is Bootstrap 4?

Bootstrap is the newest and latest version of Bootstrap. It is the most popular HTML, CSS, JavaScript framework for developing responsive, mobile first websites.

Bootstrap 3 vs. Bootstrap 4

Bootstrap 4 has some new components, faster stylesheet, more buttons, effects and more responsiveness.

Bootstrap 4 supports some the latest, stable releases of all major browsers and platforms.

Is Bootstrap Best?

Bootstrap is more than efficient to create a responsive and mobile first website but it is not the best in the industry. There is an alternative of Bootstrap named W3.CSS which is smaller, faster, and easier to use.

## SQL

SQL is the standard language for dealing with Relational Databases. SQL can be used to insert, search, update, and delete database records. SQL can do lots of other operations, including optimizing and maintenance of databases.

SQL Full Form

SQL stands for Structured Query language, pronounced as “S-Q-L” or sometimes as “See-Quel”... Relational databases like MySQL Database, Oracle, MS SQL Server, Sybase, etc. use ANSI SQL.

What is SQL used for?

Here are important reasons for using SQL

- It helps users to access data in the RDBMS system.
- It helps you to describe the data.
- It allows you to define the data in a database and manipulate that specific data.
- With the help of SQL, you can create and drop databases and tables.
- SQL offers you to use the function in a database, create a view, and stored procedure.
- You can set permissions on tables, procedures, and views.

## A Brief History of SQL

Here are important landmarks from the history of SQL:

- 1970 – Dr. Edgar F. “Ted” Codd described a relational model for databases.
- 1974 – Structured Query Language appeared.
- 1978 – IBM released a product called System/R.
- 1986 – IBM developed the prototype of a relational database, which is standardized by ANSI.
- 1989 – First ever version launched of SQL
- 1999 – SQL 3 launched with features like triggers, object-orientation, etc.
- SQL 2003 – window functions, XML-related features, etc.
- SQL 2006 – Support for XML Query Language
- SQL 2011 -improved support for temporal databases

## Types of SQL Statements

Here are five types of widely used SQL queries.

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- Data Query Language (DQL)

## List of SQL Commands

Here’s a list of some of the most commonly used SQL commands:

- CREATE – defines the database structure schema
- INSERT – inserts data into the row of a table
- UPDATE – updates data in a database
- DELETE – removes one or more rows from a table
- SELECT – selects the attribute based on the condition described by the WHERE clause
- DROP – removes tables and databases

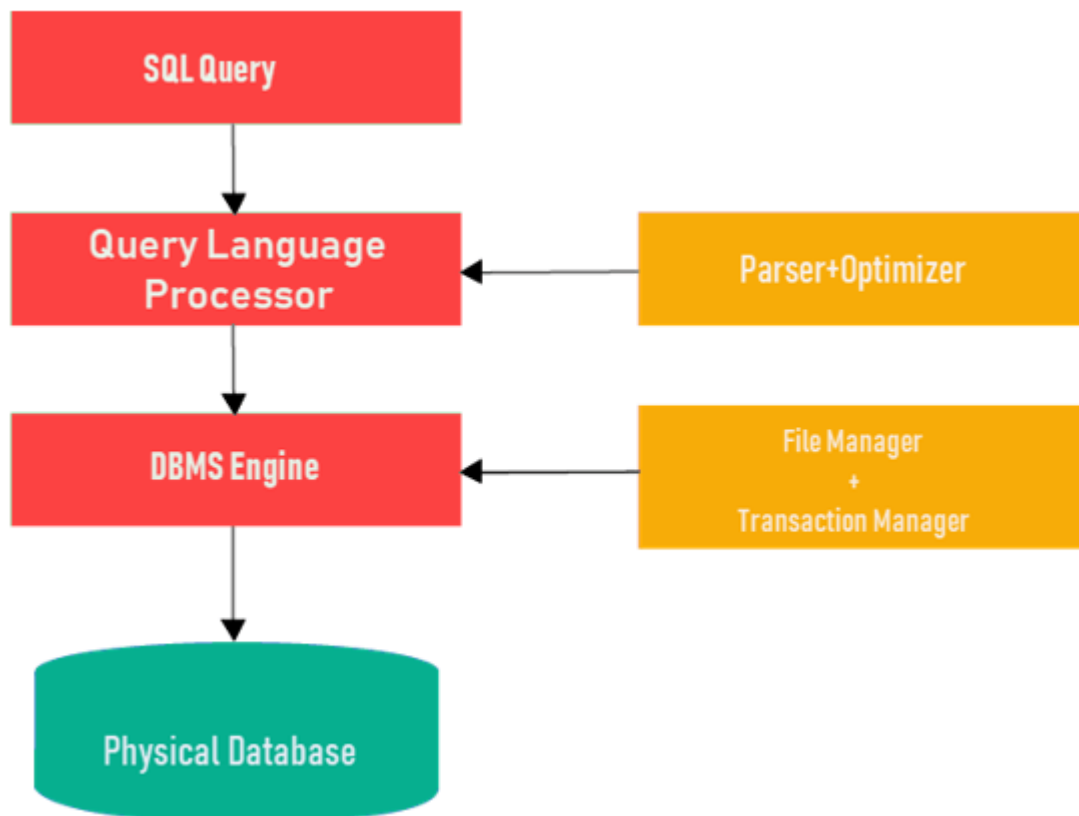
## SQL Process

When you want to execute an SQL command for any DBMS system, you need to find the best method to carry out your request, and SQL engine determines how to interpret that specific task.

Important components included in this SQL process are:

- SQL Query Engine
- Optimization Engines
- Query Dispatcher
- Classic Query Engine

A classic query engine allows you to manage all the non-SQL queries.



SQL Process

### SQL Standards

SQL is a language to operate databases. It includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI (American National Standards Institute) standard language. SQL standards are divided into several parts.

Here are some important parts of SQL standards:

Part	Description
Part 1 – SQL/Framework	It offers logical concepts.
Part 2 – SQL/Foundation	It includes central elements of SQL.
Part 3 – SQL/CLI	This standard includes central elements of SQL.
Part 4 – Persistent Stored Modules	Stored routines, external routines, and procedural language extensions to SQL.
Part 9 – Management of External Data	Adds syntax and definitions to SQL/Foundation, which allow SQL access to non-SQL data (files).
Part 10 – Object Language Bindings	Object Language Bindings: This part specifies the syntax and semantics of embedding SQL in Java™.
Part 11 – SQL/Schema	Information and Definition Schemas

Part 12 – SQL/Replication	This project began in 2000. This part helps to define the syntax and semantics to allow the of replication schemes and rules.
Part 13 – Java Routines and Type	Java Routines and Types: This part of routines using the Java Programming Language.
Part 14 – SQL/XML	SQL and XML
Part 15 – SQL/MDA	Provide SQL support for Multi-Dimensional Arrays

## SQL Language elements

Here are important elements of SQL language:

- **Keywords:** Each SQL statement contains single or multiple keywords.
- **Identifiers:** Identifiers are names of objects in the database, like user IDs, tables, and columns.
- **Strings:** Strings can be either literal strings or expressions with VARCHAR or CHAR data types.
- **Expressions:** Expressions are formed from several elements, like constants, SQL operators, column names, and subqueries.
- **Search Conditions:** Conditions are used to select a subset of the rows from a table or used to control statements like an IF statement to determine control of flow.
- **Special Values:** Special values should be used in expressions and as column defaults when building tables.
- **Variables:** Sybase IQ supports local variables, global variables, and connection-level variables.
- **Comments:** Comment is another SQL element which is used to attach explanatory text to SQL statements or blocks of statements. The database server does not execute any comment.
- **NULL Value:** Use NULL, which helps you to specify a value that is unknown, missing, or not applicable.

## What is a database in SQL?

A database is made up of a collection of tables that stores a detailed set of structured data. It is a table that contains a collection of rows, referred to as records or tuples, and columns that are also referred to as attributes.

Each column in the table is designed to store a specific type of information, for example, names, dates, dollar amounts, and numbers.

## What is NoSQL?

NoSQL is an upcoming category of Database Management Systems. Its main characteristic is its non-adherence to Relational Database Concepts. NoSQL means “Not only SQL”. The concept of NoSQL databases grew with internet giants such as Google, Facebook, Amazon etc. who deal with gigantic volumes of data.

When you use a relational database for massive volumes of data, the system starts getting slow in terms of response time. To overcome this, we could “scale up” our systems by upgrading our existing hardware. The alternative to the above problem would be to distribute our database load on multiple hosts as the load increases. This is known as “scaling out”.

NoSQL database is non-relational databases that scale-out better than relational databases and are designed with web applications in mind. They do not use SQL to query the data and do not follow strict schemas like relational models. With NoSQL, ACID (Atomicity, Consistency, Isolation, Durability) features are not guaranteed always.

## Why it makes sense to learn SQL after NoSQL?

With the advantages of NoSQL databases outlined which scale-out better than relational models, you might be thinking why one would still want to learn about the SQL database? Well, NoSQL databases are highly specialized systems and have their special usage and limitations. NoSQL suits more for those who handle huge volumes of data. The vast majority use relational databases and associated tools.

Relational databases have the following advantages over NoSQL databases.

- SQL (relational) databases have a mature data storage and management model. This is crucial for enterprise users.
- SQL database supports the notion of views that allow users to only see data that they are authorized to view. The data that they are not authorized to see is kept hidden from them.
- SQL databases support stored procedure SQL which allows database developers to implement a part of the business logic into the database.
- SQL databases have better security models compared to NoSQL databases.

The world has not deviated from the use of relational databases. There is a growing demand for professionals who can handle relational databases. Thus, learning databases and SQL basics still holds merit.

## Django

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

Django helps you write software that is:

### Complete

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

### Versatile

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc.). The site you are currently reading is built with Django!

Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

## Secure

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.

Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see Website security for more details of such attacks).

## Scalable

Django uses a component-based "shared-nothing" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

## Maintainable

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

## Portable

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavours of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

## Where did it come from?

Django was initially developed between 2003 and 2005 by a web team who were responsible for creating and maintaining newspaper websites. After creating a number of sites, the team began to factor out and reuse lots of common code and design patterns. This common code evolved into a generic web development framework, which was open-sourced as the "Django" project in July 2005.

Django has continued to grow and improve, from its first milestone release (1.0) in September 2008 through to the recently-released version 3.1 (2020). Each release has added new functionality and bug fixes, ranging from support

for new types of databases, template engines, and caching, through to the addition of "generic" view functions and classes (which reduce the amount of code that developers have to write for a number of programming tasks).

Django is now a thriving, collaborative open source project, with many thousands of users and contributors. While it does still have some features that reflect its origin, Django has evolved into a versatile framework that is capable of developing any type of website.

How popular is Django?

There isn't any readily-available and definitive measurement of popularity of server-side frameworks (although you can estimate popularity using mechanisms like counting the number of GitHub projects and StackOverflow questions for each platform). A better question is whether Django is "popular enough" to avoid the problems of unpopular platforms. Is it continuing to evolve? Can you get help if you need it? Is there an opportunity for you to get paid work if you learn Django?

Based on the number of high profile sites that use Django, the number of people contributing to the codebase, and the number of people providing both free and paid for support, then yes, Django is a popular framework!

High-profile sites that use Django include: Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic, Open Knowledge Foundation, Pinterest, and Open Stack (source: Django overview page).

Is Django opinionated?

Web frameworks often refer to themselves as "opinionated" or "unopinionated".

Opinionated frameworks are those with opinions about the "right way" to handle any particular task. They often support rapid development in a particular domain (solving problems of a particular type) because the right way to do anything is usually well-understood and well-documented. However they can be less flexible at solving problems outside their main domain, and tend to offer fewer choices for what components and approaches they can use.

Unopinionated frameworks, by contrast, have far fewer restrictions on the best way to glue components together to achieve a goal, or even what components should be used. They make it easier for developers to use the most suitable tools to complete a particular task, albeit at the cost that you need to find those components yourself.

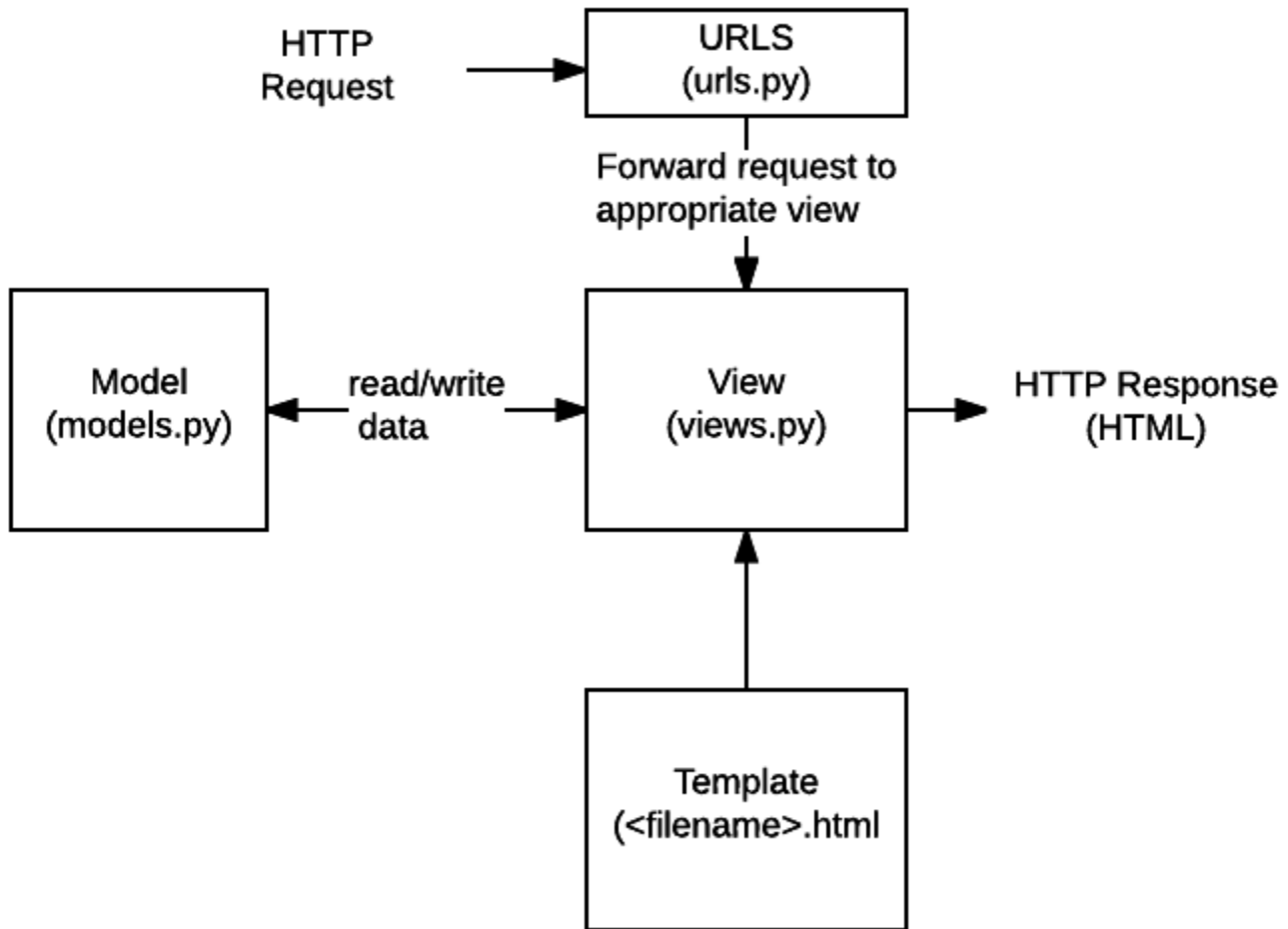
Django is "somewhat opinionated", and hence delivers the "best of both worlds". It provides a set of components to handle most web development tasks and one (or two) preferred ways to use them. However, Django's decoupled architecture means that you can usually pick and choose from a number of different options, or add support for completely new ones if desired.

What does Django code look like?

In a traditional data-driven website, a web application waits for HTTP requests from the web browser (or other client). When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from

a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser, often dynamically creating an HTML page for the browser to display by inserting the retrieved data into placeholders in an HTML template.

Django web applications typically group the code that handles each of these steps into separate files:



- **URLs:** While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.
- **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.
- **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- **Templates:** A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model



## Defining data models (models.py)

Django web applications manage and query data through Python objects referred to as models. Models define the structure of stored data, including the field types and possibly also their maximum size, default values, selection list options, help text for documentation, label text for forms, etc. The definition of the model is independent of the underlying database — you can choose one of several as part of your project settings. Once you've chosen what database you want to use, you don't need to talk to it directly at all — you just write your model structure and other code, and Django handles all the "dirty work" of communicating with the database for you.

The code snippet below shows a very simple Django model for a Team object. The Team class is derived from the django class models.Model. It defines the team name and team level as character fields and specifies a maximum number of characters to be stored for each record. The team\_level can be one of several values, so we define it as a choice field and provide a mapping between choices to be displayed and data to be stored, along with a default value.

The Django model provides a simple query API for searching the associated database. This can match against a number of fields at a time using different criteria (e.g. exact, case-insensitive, greater than, etc.), and can support complex statements (for example, you can specify a search on U11 teams that have a team name that starts with "Fr" or ends with "al").

The code snippet shows a view function (resource handler) for displaying all of our U09 teams. The list\_teams = Team.objects.filter(team\_level\_\_exact="U09") line shows how we can use the model query API to filter for all records where the team\_level field has exactly the text 'U09' (note how this criteria is passed to the filter() function as an argument, with the field name and match type separated by a double underscore: team\_level\_\_exact).

## Rendering data (HTML templates)

Template systems allow you to specify the structure of an output document, using placeholders for data that will be filled in when a page is generated. Templates are often used to create HTML, but can also create other types of document. Django supports both its native templating system and another popular Python library called Jinja2 out of the box (it can also be made to support other systems if needed).

The code snippet shows what the HTML template called by the render() function in the previous section might look like. This template has been written under the assumption that it will have access to a list variable called youngest\_teams when it is rendered (this is contained in the context variable inside the render() function above). Inside the HTML skeleton we have an expression that first checks if the youngest\_teams variable exists, and then iterates it in a for loop. On each iteration the template displays each team's team\_name value in an <li> element.

## How data is stored

- Computers represent data, including video, images, sounds and text, as binary values using patterns of just two numbers: 1 and 0. A bit is the smallest unit of data, and represents just a single value. A byte is eight binary digits long. Storage and memory is measured in megabytes and gigabytes.

- The units of data measurement continue to grow as the amount of data collected and stored grows. The relatively new term "brontobyte," for example, is data storage that is equal to 10 to the 27th power of bytes.
- Data can be stored in file formats, as in mainframe systems using ISAM and VSAM. Other file formats for data storage, conversion and processing include comma-separated values. These formats continued to find uses across a variety of machine types, even as more structured-data-oriented approaches gained footing in corporate computing.
- Greater specialization developed as database, database management system and then relational database technology arose to organize information.

### **Types of data**

Growth of the web and smartphones over the past decade led to a surge in digital data creation. Data now includes text, audio and video information, as well as log and web activity records. Much of that is unstructured data.

The term big data has been used to describe data in the petabyte range or larger. A shorthand take depicts big data with 3Vs -- volume, variety and velocity. As web-based e-commerce has spread, big data-driven business models have evolved which treat data as an asset in itself. Such trends have also spawned greater preoccupation with the social uses of data and data privacy.

Data has meaning beyond its use in computing applications oriented toward data processing. For example, in electronic component interconnection and network communication, the term data is often distinguished from "control information," "control bits," and similar terms to identify the main content of a transmission unit. Moreover, in science, the term data is used to describe a gathered body of facts. That is also the case in fields such as finance, marketing, demographics and health.

### **Data management and use**

With the proliferation of data in organizations, added emphasis has been placed on ensuring data quality by reducing duplication and guaranteeing the most accurate, current records are used. The many steps involved with modern data management include data cleansing, as well as extract, transform and load (ETL) processes for integrating data. Data for processing has come to be complemented by metadata, sometimes referred to as "data about data," that helps administrators and users understand database and other data.

Analytics that combine structured and unstructured data have become useful, as organizations seek to capitalize on such information. Systems for such analytics increasingly strive for real-time performance, so they are built to handle incoming data consumed at high ingestion rates, and to process data streams for immediate use in operations.

Over time, the idea of the database for operations and transactions has been extended to the database for reporting and predictive data analytics. A chief example is the data warehouse, which is optimized to process questions about operations for business analysts and business leaders. Increasing emphasis on finding patterns and predicting business outcomes has led to the development of data mining techniques.

### **Data professionals**

The database administrator profession is an offshoot of IT. These database experts work on designing, tuning and maintaining the database.

The data profession took firm root as the relational database management system (RDBMS) gained wide use in corporations, beginning in the 1980s. The relational database's rise was enabled in part by the Structured Query Language (SQL). Later, non-SQL databases, known as NoSQL databases, arose as an alternative to established RDBMSes.

Today, companies employ data management professionals or assign workers the role of data stewardship, which involves carrying out data usage and security policies as outlined in data governance initiatives.

A distinct title -- the data scientist -- has appeared to describe professionals focused on data mining and analysis. The benefit of presenting data science in an evocative manner has even given rise to the data artist; that is, an individual adept at graphing and visualizing data in creative ways.

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.

### **What is the difference between data cleaning and data transformation?**

Data cleaning is the process that removes data that does not belong in your dataset. Data transformation is the process of converting data from one format or structure into another. Transformation processes can also be

referred to as data wrangling, or data munging, transforming and mapping data from one "raw" data form into another format for warehousing and analyzing. This article focuses on the processes of cleaning that data.

## **How do you clean data?**

While the techniques used for data cleaning may vary according to the types of data your company stores, you can follow these basic steps to map out a framework for your organization.

### **Step 1: Remove duplicate or irrelevant observations**

Remove unwanted observations from your dataset, including duplicate observations or irrelevant observations. Duplicate observations will happen most often during data collection. When you combine data sets from multiple places, scrape data, or receive data from clients or multiple departments, there are opportunities to create duplicate data. De-duplication is one of the largest areas to be considered in this process. Irrelevant observations are when you notice observations that do not fit into the specific problem you are trying to analyze. For example, if you want to analyze data regarding millennial customers, but your dataset includes older generations, you might remove those irrelevant observations. This can make analysis more efficient and minimize distraction from your primary target—as well as creating a more manageable and more performant dataset.

### **Step 2: Fix structural errors**

Structural errors are when you measure or transfer data and notice strange naming conventions, typos, or incorrect capitalization. These inconsistencies can cause mislabeled categories or classes. For example, you may find “N/A” and “Not Applicable” both appear, but they should be analyzed as the same category.

### **Step 3: Filter unwanted outliers**

Often, there will be one-off observations where, at a glance, they do not appear to fit within the data you are analyzing. If you have a legitimate reason to remove an outlier, like improper data-entry, doing so will help the performance of the data you are working with. However, sometimes it is the appearance of an outlier that will prove a theory you are working on. Remember: just because an outlier exists, doesn't mean it is incorrect. This step is needed to determine the validity of that number. If an outlier proves to be irrelevant for analysis or is a mistake, consider removing it.

### **Step 4: Handle missing data**

You can't ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered.

1. As a first option, you can drop observations that have missing values, but doing this will drop or lose information, so be mindful of this before you remove it.
2. As a second option, you can input missing values based on other observations; again, there is an opportunity to lose integrity of the data because you may be operating from assumptions and not actual observations.
3. As a third option, you might alter the way the data is used to effectively navigate null values.

### **Step 5: Validate and QA**

At the end of the data cleaning process, you should be able to answer these questions as a part of basic validation:

- Does the data make sense?
- Does the data follow the appropriate rules for its field?
- Does it prove or disprove your working theory, or bring any insight to light?
- Can you find trends in the data to help you form your next theory?
- If not, is that because of a data quality issue?

False conclusions because of incorrect or “dirty” data can inform poor business strategy and decision-making. False conclusions can lead to an embarrassing moment in a reporting meeting when you realize your data doesn’t stand up to scrutiny. Before you get there, it is important to create a culture of quality data in your organization. To do this, you should document the tools you might use to create this culture and what data quality means to you.

### **Components of quality data**

Determining the quality of data requires an examination of its characteristics, then weighing those characteristics according to what is most important to your organization and the application(s) for which they will be used.

5 characteristics of quality data

1. **Validity.** The degree to which your data conforms to defined business rules or constraints.
2. **Accuracy.** Ensure your data is close to the true values.
3. **Completeness.** The degree to which all required data is known.
4. **Consistency.** Ensure your data is consistent within the same dataset and/or across multiple data sets.
5. **Uniformity.** The degree to which the data is specified using the same unit of measure.

### ***Benefits***

Make database management more Python-like

Starting a Django project allows you to build your application’s entire data model in Python without needing to use SQL. Using an object-relational mapper (ORM), Django converts traditional database structure into Python classes to make it easier to work within a fully Python environment. Django-MySQL supports the JSON data type and related functions.

In Django, your database tables become Python classes. Web applications access and manage data through Django models. The fields of the database are simply converted into class attributes. If you’re familiar with class attribute definition in Python, you can easily design and manage a Django database.

Django Web Framework offers a shortcut to full integration with your application's database. It provides CRUD (create, read, update, delete) functionality, HttpResponseRedirect and cross-site scripting, supplies user management capabilities, offers software administration features and more. You import the packages, connect to your database and then get back to work developing the parts of your application that make your product unique.

### Create dynamic pages with templates

Because Django is designed to be used for web app development, it needs a way to easily create dynamic HTML that displays your user's unique data. The Django application produces that dynamic HTML with a built-in templating engine called the Django template language (DTL).

An HTML template allows Django developers to combine static elements (including design elements such as colors, logos, or text) with data (such as user names or locations) to create a new web page on the fly. With model-view-controller (MVC), if you want your application to greet a user by name when they log in, you can build a template that displays the static text ("Welcome to the site, X") then use a dynamic placeholder to automatically display the user's first name from your database. When the page renders, it will combine the dynamic elements with the static ones to create a seamless user experience.

### Enhance security

When answering, "What is Django", we must talk about what special features Django offers for security. Web apps are frequent targets of hackers, especially applications that store user login information or financial data. Django offers features to help protect your application and your users.

One of the biggest risks for sites that accept user-entered data is that a malicious user will inject code with their data that can have a disastrous effect on your system. To protect against attacks like these, Django templates automatically escape common HTML characters in any user-entered field. For example, it will automatically convert '<' to '&lt;' to make it difficult to inject malicious code into your program. Django protects from SQL injection in a similar way, reinterpreting unauthorized commands so that users can't sneak their own code into your database.

Web developers can also count on Django APIs to automatically use cross-site request forgery (CSRF) protection to insert user-specific secret tokens into POST requests. As a result, web developers can prevent malicious users from duplicating other POST requests to masquerade as authorized users.

The protection of Django goes beyond its explicit security features: security efforts are enhanced by the extensive experience and expertise of the Django user base. If you build your entire web app from scratch, you run the risk of accidentally introducing a security vulnerability into your module. Django packages are widely used, open source and well reviewed by web developers, so you can be more confident that they'll protect your data.

### *Scaling Django*

One of the biggest challenges in web development is scalability. Your application must handle exactly the number of active connections that are required at any given time. If the connections exceed your estimate, your users will experience lags and downtime. Estimate too conservatively and you'll pay for bandwidth and web servers you're not using. You need an application that can grow as you gain more users.

Django web framework makes scaling easy. Because a Django app can manage your user sessions, you can add more instances of your application and transfer the user's experience across the instances without losing data.

Many Django project developers also use a cache manager such as Varnish to pre-load the static elements of the site for users.

You'll need to carefully configure the cache manager to avoid accidentally caching your entire site, including the dynamic elements. A poorly configured cache could also lead to sharing data from one user to the next. You can avoid over-caching by using Django's native (but limited) cache framework in conjunction with an external cache manager to free up performance resources while being selective on what is stored.

When answering the question, "What is Django," one answer is that it is a highly scalable web framework. Just how scalable is it? Instagram and Disqus are among the very large sites that use Django to support their huge user bases.

This is how websites – even simple ones designed by a single person – can still include advanced functionality like authentication support, management and admin panels, contact forms, comment boxes, file upload support, and more. In other words, if you were creating a website from scratch you would need to develop these components yourself. By using a framework instead, these components are already built, you just need to configure them properly to match your site.

The official project site describes Django as “a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.”

Django offers a big collection of modules which you can use in your own projects. Primarily, frameworks exist to save developers a lot of wasted time and headaches and Django is no different.

You might also be interested in learning that Django was created with front-end developers in mind. “Django's template language is designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed.”

If you're going to be working with Python, especially for web applications or web design, you'll want to remember the Django framework. It will certainly come in handy.

[CherryPy](#) is another Python-based framework that is great to work with, although it is designed with the absolute minimalist in mind. It's a framework you'll want to explore after you already have some experience working with Python.

## Chapter V. IMPLEMENTATION

### *A. Technologies Used*

Various front-end and back-end technologies are available in this era of digitalization. The technologies used in this project are discussed briefly in the following sections. 1) *Front End Technologies*

#### ***HTML***

HyperText Markup Language (HTML) is the set of markup symbols or codes inserted into a file intended for display on the Internet. The markup tells web browsers how to display a web page's words and images.

Each individual piece markup code (which would fall between "<" and ">" characters) is referred to as an element, though many people also refer to it as a tag. Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

Hypertext Mark-up Language is the computer language that facilitates website creation. The language, which has code words and syntax just like any other language, is relatively easy to comprehend and, as time goes on, increasingly powerful in what it allows someone to create. HTML continues to evolve to meet the demands and requirements of the Internet under the guise of the World Wide Web Consortium, the organization that designs and maintains the language; for instance, with the transition to Web 2.0.

Hypertext is the method by which Internet users navigate the web. By clicking on special text called hyperlinks, users are brought to new pages. The use of hyper means it is not linear, so users can go anywhere on the Internet simply by clicking on the available links. Markup is what HTML tags do to the text inside of them; they mark it as a specific type of text. For example, mark-up text could come in the form of boldface or italicized type to draw specific attention to a word or phrase.

At its core, HTML is a series of short codes typed into a text-file. These are the tags that power HTML's capabilities. The text is saved as an HTML file and viewed through a web browser. The browser reads the file and translates the text into a visible form, as directed by the codes the author used to write what becomes the visible rendering. Writing HTML requires tags to be used correctly to create the author's vision.

The tags are what separate normal text from HTML code. Tags are the words between what are known as angle-brackets, which allow graphics, images, and tables to appear on the webpage. Different tags perform different functions. The most basic tags apply formatting to text. As web interfaces need to become more dynamic, Cascading Style Sheets (CSS) and JavaScript applications may be used. CSS makes web pages more accessible and JavaScript adds power to basic HTML.



## CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

- Create Stunning Web site - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
- Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- Control web - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
- Learn other languages - Once you understand the basic of HTML and CSS then other related technologies like JavaScript, php, or angular are become easier to understand

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

## *JavaScript/JQuery*

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as Live Script, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name Live Script. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The [ECMA-262 Specification](#) defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

jQuery is a fast, small, cross-platform and feature-rich JavaScript library. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers.

The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.

jQuery is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as write less do more because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

- jQuery is a small, fast and lightweight JavaScript library.
- jQuery is platform-independent.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.

### ***BootStrap***

Bootstrap (Shenoy & Sossou, 2014) is a free and opensource CSS framework directed at responsive, MobileFirst front-end web development. It contains CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. To use bootstrap, we are required to either install in our system or use CDN. CDN is short for content delivery network. A CDN is a system of distributes servers that deliver pages and other web content to a user, based on the geographic locations of the user, the origin of the webpage and the content delivery server.

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.

- It facilitates you to create responsive designs.



## 2) Back End Technologies

### a) *Python*

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991 (Kuhlman, 2011), Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects.

In this website, python is used as backend language to code database part and all functionalities that the website can perform. The version of Python used in this development is Python 3.6.

### b) *Django*

Django (Holovaty & Kaplan-Moss, 2008) is a highlevel Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so we can focus on writing our app without needing to reinvent the wheel. It's free and open source. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code; low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models. The version of Django used during development is Django 2.1.5.

### c) *SQLite*

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. By default, Django used SQLite3 as its default database. Django provides a specific way to define our database using the python programming language.

d) *Jinja2*

Jinja2 (Lokhande et al, 2015) is one of the most used template engines for Python. It is inspired by Django's templating system but extends it with an expressive language that gives template authors a more powerful set of tools.

It adds Sandboxed execution mode i.e. every aspect of the template execution is monitored and explicitly whitelisted or blacklisted, whatever is preferred. Some features of Jinja2 are enlisted below:

- Powerful automatic HTML escaping system for cross-site scripting prevention.
- Template inheritance makes it possible to use the same or a similar layout for all templates.
- Optional ahead-of-time compilation and Configurable syntax i.e. we can reconfigure Jinja2 to better fit output formats such as LaTeX or JavaScript.

The version of Jinja2 used as a templating language in this project is 2.10.

*B. Hardware and Software Requirements*

The project developed satisfies all the functional and nonfunctional requirements enlisted. The following specifications are required for the project to run on any device.

*1) System Specifications*

Processor: Intel(R) Core (TM) i5-5005U CPU @ 2.00GHz

RAM: 2 GB

System Type: 32-bit/64-bit operating system, x32 or x64 based processor

Operating System: Windows 7/8/10.

*2) Software Interface*

Front End: HTML, CSS, Bootstrap, JQuery

Backend: Django

Local Access Link: localhost:8000

Global Access Link: <https://www.shreic.com/home/>

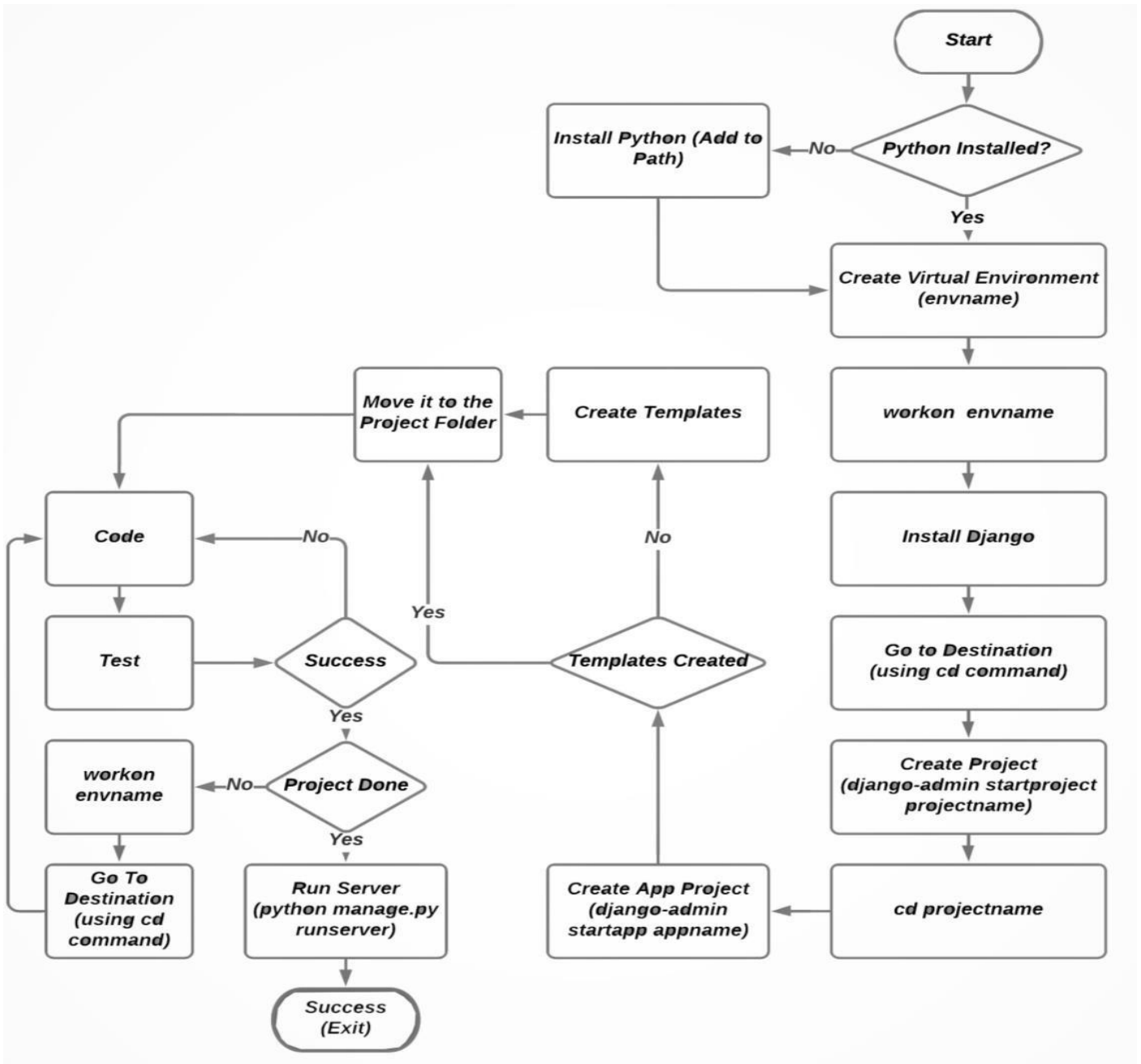
*C. Methodology*

The Project is developed via multiple steps. The major steps are enlisted here:

1. Installing Python and adding it to the windows path.
2. Creation of Virtual Environment (Following commands are written in Command Prompt)
  - pip install virtualenvwrapper-win
  - mkvirtualenvironmentname (any name can be given)
  - workonenvironmentname
3. Installing Django
  - pip install Django

4. Go to Destination Place where you want the project to be kept, using cd command.
5. Create Project as follows
  - `django-admin startproject someprojectname`
  - `cd someprojectname`
6. Create App of the project as
  - `django-admin startapp appname`
  - `python manage.py makemigrations`
  - `python manage.py migrate`
7. Copy the Template Folder (if Front End Template is downloaded) to the project folder created.
8. Run Server (localhost:8000)
  - `python manage.py runserver`

The Flow Chart illustrates the steps that are required in order to install the prerequisites of the project and then the steps involved in the development of project. Commands written in bracket are to be run on Command Prompt. They are the steps that are required to install virtual environment and to run local server on the system project is to be developed on. The code of backend and frontend can be coded in any code editor (Sublime Text was used in this project).



All the changes that were made in the project can be seen on the local server. The Data was stored on online cloud service Cloudinary.

## Chapter VI. RESULT

Few snaps of the website are shown here showing the main functionalities of the project.

The Home Page has the top bar consisting of the website logo, the drop-down menu of categories, sell books button, add notes button, request an educational resource button, a dynamic search bar and the signup/login option. Below the top bar is the banner representing the main motive of the website. After that there is stats of the total number of books, notes and users of the website. On the bottom right corner is the recent chat button which shows the recent chats the user had done.

The dynamic search bar developed for simplifying the searching option for users of the website. A dynamic search bar possesses the feature of typing suggestions and helps the user to autocomplete the searching process. A form is created in HTML code of the home page and a function is coded in the views.py page that retrieves all books and notes uploaded, filters it and return the names on the basis of letters typed in the search box and an URL is written in urls.py to refer to the function created.

An Add a book form is created for adding a book to sell or donate through the website. On clicking the Sell Button on the top bar following form will open. The form consists of Book Name, Dropdown menu for Category and Subject, Author Name, Publication Name and Year and MRP, Description and the Cover Image of the book. On Submit, the record will be added to Book Details Model created in models.py. And the form created for uploading notes (question papers/class notes/syllabus/ebooks) on the website. On clicking the Add Button on the top bar following form will open. The form consists of Notes Name, Dropdown menu for Notes Type, Section, Category and Subject, Publication Name and Year, Description and the file (pdf format). On Submit, the record will be added to Notes Details Model created in models.py.

The dropdown menu that appears on the screen when the mouse is hovered over the Category Button shows all the Categories created and their subcategories. The dropdown menu is divided into categories and each category consists of different subcategories. These Categories and their corresponding Subcategories are retrieved with the help of functions written in views.py and called in the HTML page.

The Computer Science subcategory of Science under Academics Section (Academics/Science/Computer Science) is a sample for a Particular Subcategory Page. The side bar consists of filters based on the type of notes. Function selected products in views.py filters the notes and books of the selected category with the help of ID of the subcategory created using get () and filter ().illustrates the chat-box created for chatting between the User and the Seller of a Book. By the help of chat and mutual understanding the transaction can be done successfully between the two. The sidebar shows the persons the user recently chatted to and the right side shows chat of the particular person the user is chatting to.

Study Material Hub

Not secure | studymaterialhub.ga/index.php

All Notes B-Tech BCA BBA MCA Upload Project

**B-Tech CSE Notes** ●  
05 Aug 2021  
Please Click to Download Notes

**B-Tech Macheanical No...**  
05 Aug 2021  
Please Click to Download Notes

**B-Tech Electrical Notes...**  
07 Aug 2021  
Please Click to Download Notes

**B-Tech Civil Notes** ●  
07 Aug 2021  
Please Click to Download Notes

**BBA Notes** ●  
05 Aug 2021  
Please Click to Download Notes

**BCA Notes** ●  
07 Aug 2021  
Please Click to Download Notes

**MCA Notes** ●  
07 Aug 2021  
Please Click to Download Notes

Type here to search

23°C Haze 11:57 AM 02-12-2021

Study Material Hub

Not secure | studymaterialhub.ga/btechcse.php

All Semester Sem - 1 Sem-2 Sem-3 Sem-4 Sem-5 Sem-6 Sem-7 Sem-8 Home

**Universal Human Valu...**  
05 Aug 2021  
Please Click to Download Notes  
Download Module ▾

**Psychology and Sociol...**  
05 Aug 2021  
Please Click to Download Notes  
Download Module ▾

**Matrices And Differenti...**  
05 Aug 2021  
Please Click to Download Notes  
Download Module ▾

**Energy and Environme...**  
05 Aug 2021  
Please Click to Download Notes  
Download Module ▾

**Data Structure using C** ●  
07 Aug 2021  
Please Click to Download Notes  
Download Module ▾

**Theory of Automata a...**  
07 Aug 2021  
Please Click to Download Notes  
Download Module ▾

**Digital Design and co...**  
07 Aug 2021

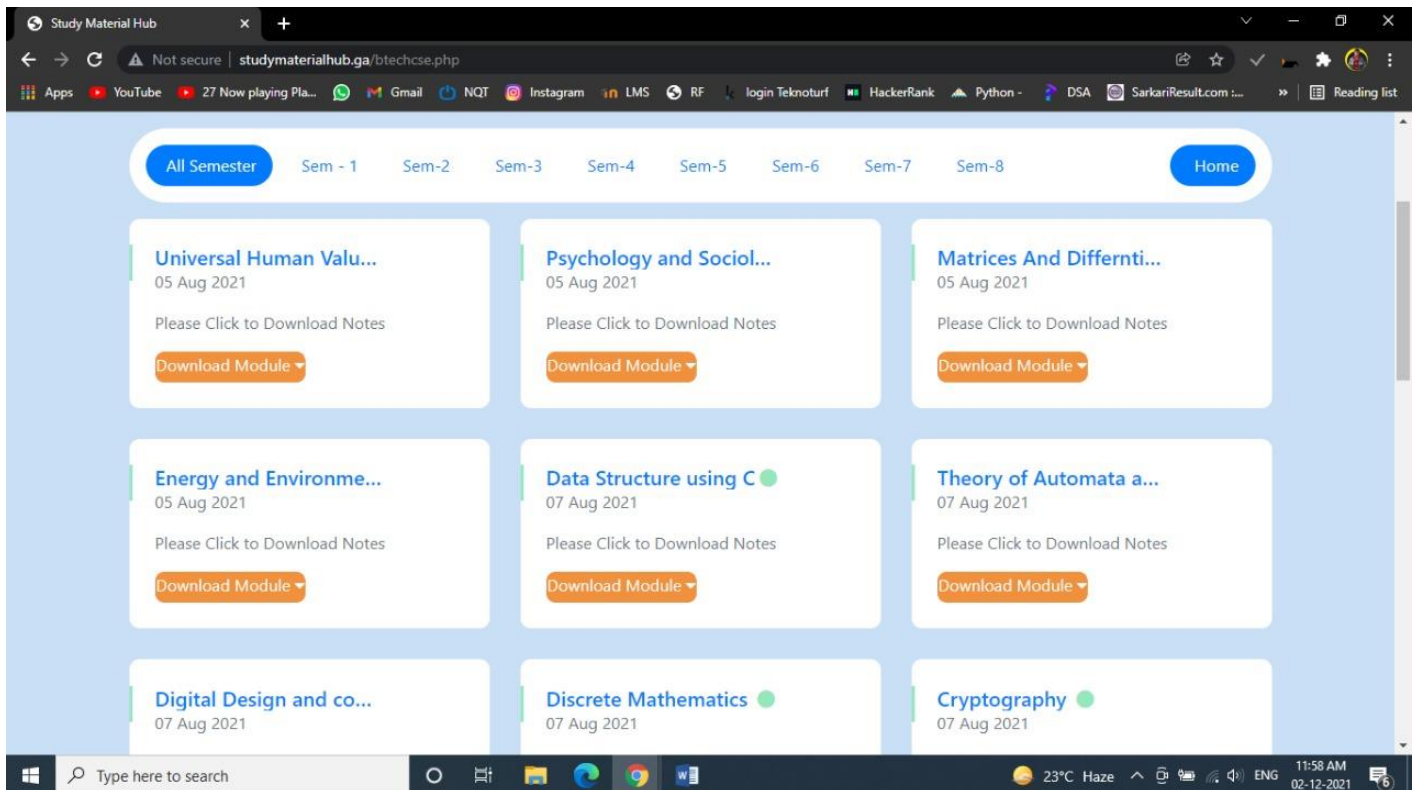
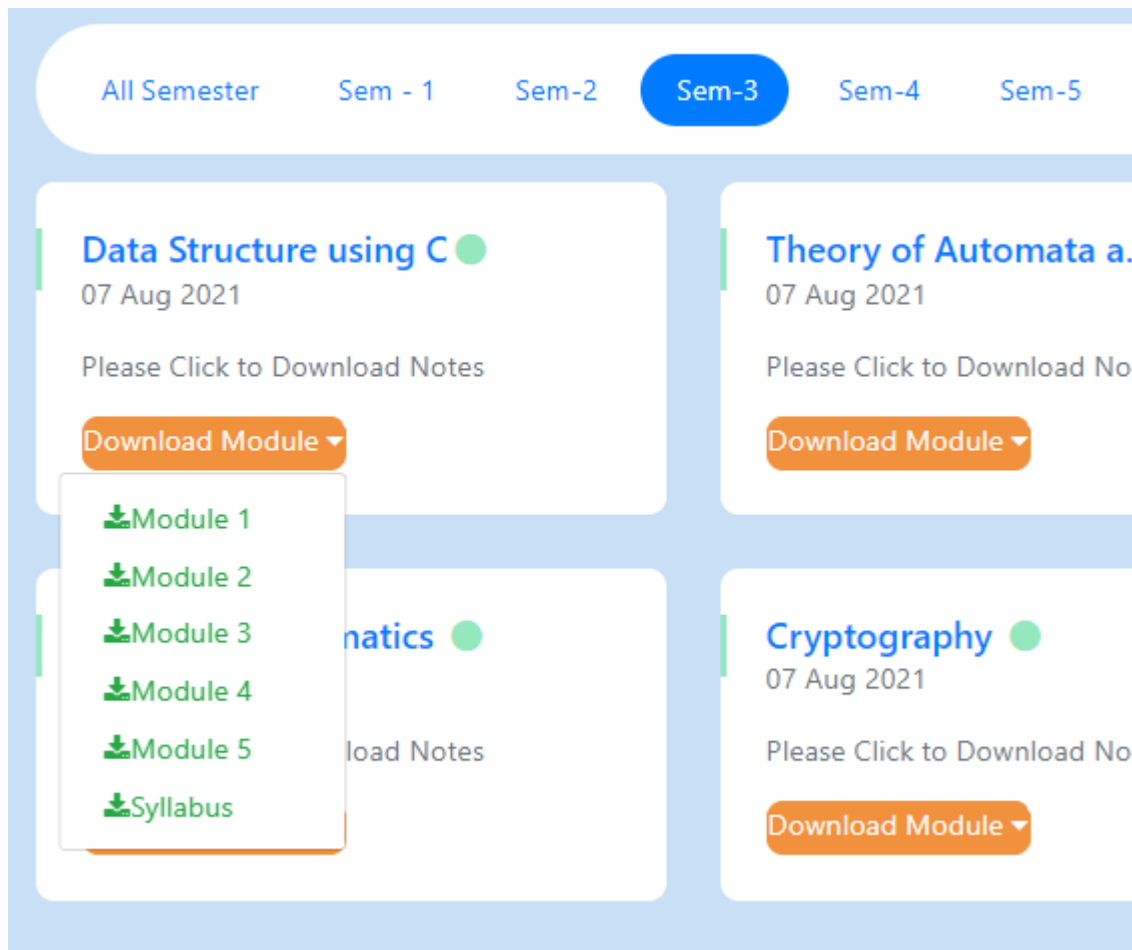
**Discrete Mathematics** ●  
07 Aug 2021

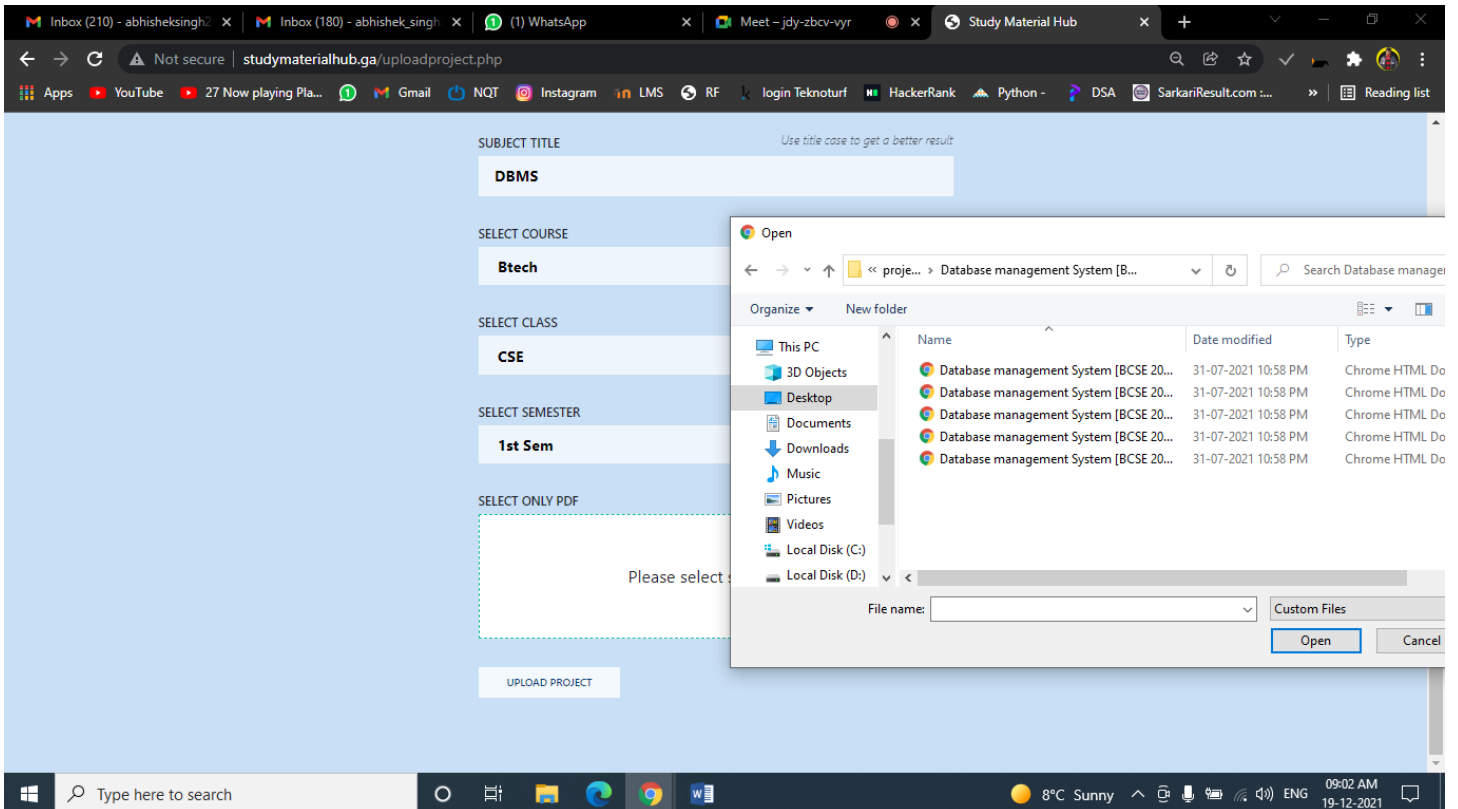
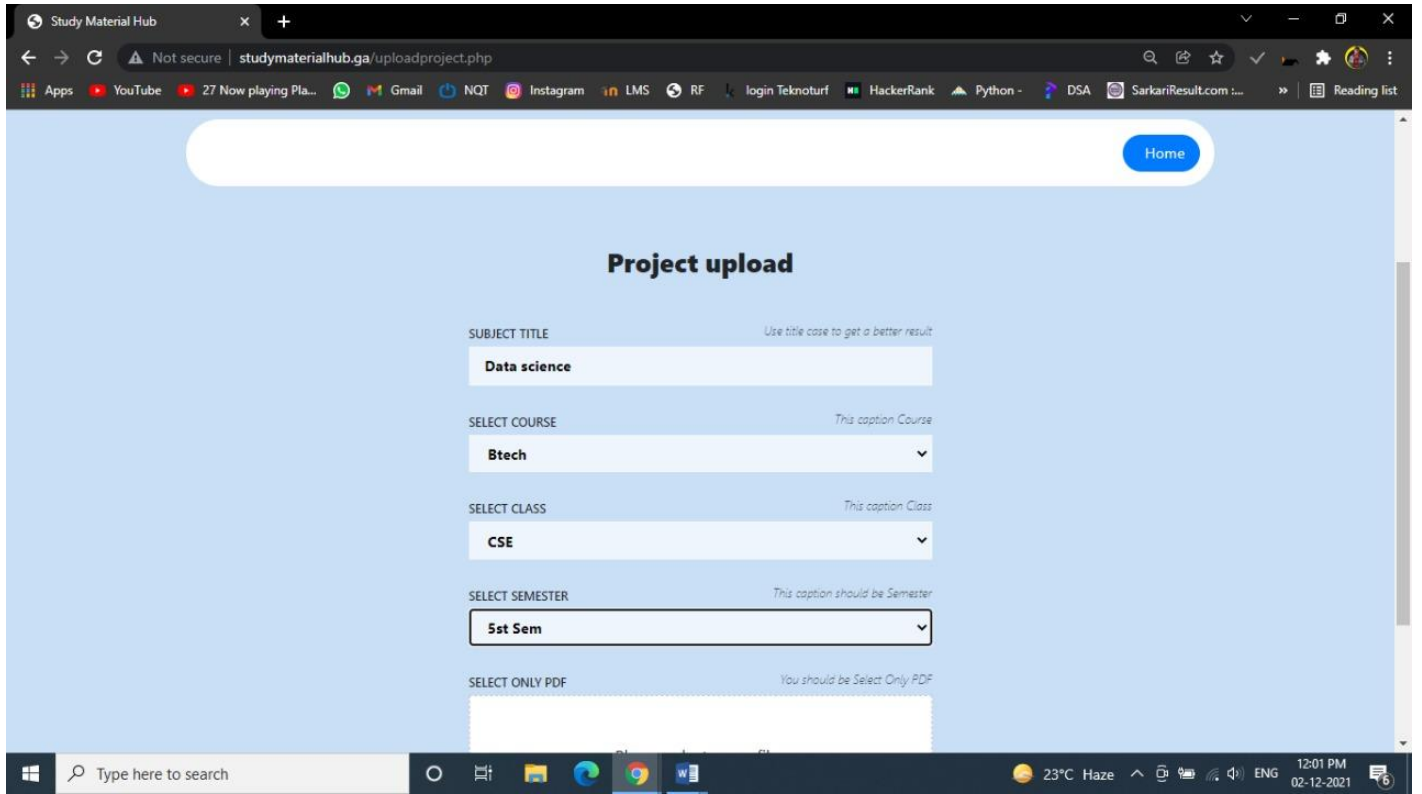
**Cryptography** ●  
07 Aug 2021

Type here to search

23°C Haze 11:58 AM 02-12-2021







## VII. TESTING

In this project the testing (Thakur, 2017) has been done as follows:

### A. *Black Box Testing*

Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester.

In this testing process, ten Users were selected. Firstly, they were said to Sign Up on the website. After successful registration of each user, they were asked to upload notes (the samples were given) in pdf format in which 8 of them succeeded. Two of the Users were unable to upload the notes. This issue was resolved and then further process took place. Following the note uploading testing, each User was provided an old Book, which they uploaded on the server to test the functionality of the book adding feature.

Further, the users were asked to fill the “request a note” form where the user can request a book or note to be made available on the website. After that, the users were said to chat with each other to check the simplicity and functionality of the chat-box which was found satisfiable.

At last, in the process of Black Box Testing, Users were requested to review the whole website and each functionality and their feedback was recorded. According to which, corrections and better implementations were made on the website.

### B. *White Box Testing*

White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

The White Box Testing is done by the developers only. There are many kinds of White Box Testing. In this project, Unit Testing was done i.e. each unit of code was separately tested and was integrated lastly. The testing of source code involved

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops
- Testing of each statement, object, and function on an individual basis.

Different test cases were made for each unit of source code and were tested. For each test case, the desired output was expected. When the desired output was not encountered it led to the bug. Each error was removed from the source code and all units were integrated at last. The benefit of unit testing was that errors/bugs were identified at the initial level of development thus avoiding any big error to occur at a further level of development.

Also, in the testing process, the security issues have been resolved. The CSRF middleware and template tag in Django provides easy-to-use protection against Cross-Site Request Forgeries. This type of attack occurs when a malicious website contains a link, a form button or some JavaScript that is intended to perform some action on your website, using the credentials of a logged-in user who visits the malicious site in their browser. A related type of attack, ‘login CSRF’, where an attacking site tricks a user’s browser into logging into a site with someone else’s credentials, is also covered. The CSRF middleware is activated by default in the middleware setting in the Django framework. Ajax Forms are used in the project and each form is tagged with the token to secure the forms

from unwanted inputs that can corrupt the database. The web application is made to exchange data with the webserver securely by deploying the web application behind HTTPS.

## CONCLUSION

After analyzing the results obtained, the project developed can be considered satisfiable. It can be concluded that the website will be very helpful to students in their educational life as it provides all educational resources required in a college or school life. As the project works as an Educational cum E-Commerce Website and thus students can donate or sell their old books too.

To conclude, the project is developed using the proper Software Engineering process, following the Iterative Model of SDLC. A Project Control List was created after doing the feasibility study for functionalities as well as non-functional requirements. Then proper schema and tables that were supposed to be required in the development process were made and relationships between each table were drawn. For this ER Diagram was made which has been illustrated in the paper. Also, the flow chart was created so that each process can be done sequentially. After that, each task from the project control list was coded, tested using White Box Testing and implemented separately as per the Iterative Model. At last every unit was integrated and users were selected for Black Box Testing. Each user was asked to run the project and test each functionality of the project. After the testing, feedback and suggestions were recorded and accordingly the amendments were made. Security issues were resolved with the help of CSRF tags given by the Django Framework and by deploying the Web Application behind HTTPS.

The approach used in the System Development Model can act as a roadmap for the development of similar kinds of Web Applications efficiently.

Also, for future works few more features can be added to the project. Some of them that have been enlisted are using the platform as an online assignment submission platform, creation of a chatroom consisting of teachers and students of particular university/college and adding digital payment methods for easier transactions.

## REFERENCES

1. Buy Second Hand Books, Old Books, Used Books Online in India. (n.d.). Retrieved from <https://www.usedbooksfactory.com/>.
2. Viorica-Torii, C., & Carmen, A. (2013). The Impact of Educational Technology on the Learning Styles of Students. *Procedia-Social and Behavioral Sciences*, 851-855.
3. About Company. (n.d.). Retrieved from <http://www.tutorialspoint.com/>.
4. The Physics Classroom. (n.d.). Retrieved from <https://www.physicsclassroom.com/>. Kelly, L., & Breault, K. (2006). Developing Educational Websites: Investigating Internet Use by Students and Teachers. In *Proceedings of Thinking, Evaluating, Rethinking, ICOM-CECA Conference*, Rome.
5. AglaSem Admission. (n.d.). Retrieved from <https://admission.aglasem.com/>.
6. Sachan, N. (2019, February 20). Welcome to BHU Student Club, BHU Student Club. Retrieved from <http://bhustudentclub.in/>.
7. Jalote, P. (2003). *An Integrated Approach towards Software Engineering*. Narosa Publishing House.
8. Musciano, C., & Kennedy, B. (1996). *HTML, The Definitive Guide*. O'Reilly & Associates.
9. Powell, T. A. (2010). *HTML & CSS: The Complete Reference*. The McGraw-Hill Companies.
10. Flanagan, D. (2006). *JavaScript: The Definitive Guide*. O'Reilly Media, Inc. Shenoy, A., & Sossou, U. (2014). *Learning Bootstrap*. Packt Publishing Ltd.
11. Kuhlman, D. (2011). *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. Platypus Global Media.
12. Holovaty, A., & Kaplan-Moss, J. (2008). *The Definitive Guide to Django: Web Development done right*. Apress.
13. Lokhande, P. S., Aslam, F., Hawa, N., Munir, J., & Gulamgaus, M. (2015). Efficient way of Web Development using Python and Flask. *International Journal of Advanced Research in Computer Science*, 54-57.
14. Thakur, M. S. (2017). Review on Structural Software Testing Coverage Approaches. *International Journal of Advance Research, Ideas and Innovations in Technology*, 281-286.

\*\*\*