



Rayat Shikshan Sanstha's

**Project Report on**  
**TEXT TO HTML CONVERTER**

*For*

S.M.Joshi College, Hadapsar ,Pune 28.

*Submitted To/*

**Department of Master of Computer**  
**ScienceSavitribai Phule Pune University,**  
**Pune**

In the partial fulfillment of the award of degree

***Master Of Computer Science( CS )***

***Submitted by,***

***PRATIK KAILAS BARGUJE***

Under the guidance of

***( Prof. Yadav Sangeeta)***

Through,

The Head Department of Master of Computer Science(CS)

**Pune University, Pune**

"Education Through Self Help is Our **Motto**"  
2022 - 23

**Year**



Rayat Shikshan Sanstha's  
**S.M.Joshi College, Hadapsar**

**CERTIFICATE**

*This is to certify that,*

**Mr. PRATIK KAILAS BARGUJE** *has*  
*completed her work on project report*  
*entitled*

**“TEXT TO HTML CONVERTER for the academic  
year of 2022-2023.”**

*for S.M.Joshi College, Hadapsar, Pune 28 in Visual*

*6.0 in the partial fulfillment of the award of the degree  
“Master of Computer Science”(MCS–Sem-3) through  
Savitribai Phule Pune University, Pune.*

*This work has been carried out under my guidance to the  
best of my knowledge and belief the matter presented in this /  
project report has not been submitted earlier and it is his  
original work.*

**HEAD OF DEPARTMENT**

**EXTERNAL EXAMINER**

**PROJECT GUID**

**Principal**

# **INTRODUCTION TO PROJECT**

## **TEXT TO HTML**

- **INTRODUCTION**
- **EXISTING SYSTEM**
- **NEED AND SCOOP OF COMPUTER SYSTEM**

# INTRODUCTION

This Project will give simple approach with help of this project we can convert TEXT to HTML. With the help of this project NON BCS OR MCS student will be able to write code in Text To HTML. You can create this Tags <h1>HELLO WORLD</h1> just write in text like this first select the heading1 and write a HELLO WORLD it will be convert into HTML

Generally, the HTML Language are written in Code , HTML language cannot be write normal language, Like (English,Hindi,French ). The **HyperText Markup Language** or **HTML** is the standard [markup language](#) for documents designed to be displayed in a [web browser](#). It can be assisted by technologies such as [Cascading Style Sheets](#) (CSS) and [scripting languages](#) such as [JavaScript](#).

[Web browsers](#) receive HTML documents from a [web server](#) or from local storage and [render](#) the documents into multimedia web pages. HTML describes the structure of a [web page semantically](#) and originally included cues for the appearance of the document.

[HTML elements](#) are the building blocks of HTML pages. With HTML constructs, [images](#) and other objects such as [interactive forms](#) may be embedded into the rendered page. HTML provides a means to create [structured documents](#) by denoting structural [semantics](#) for text such as headings, paragraphs, lists, [links](#), quotes, and other items. HTML elements are delineated by *tags*, written using [angle brackets](#).

Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

In 1980, [physicist Tim Berners-Lee](#), a contractor at [CERN](#), proposed and prototyped [ENQUIRE](#), a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an [Internet](#)-based [hypertext](#) system.<sup>[3]</sup> Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN [data systems](#) engineer [Robert Cailliau](#) collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes<sup>[4]</sup> from 1990 he listed<sup>[5]</sup> "some of the many areas in which hypertext is used" and put an [encyclopedia](#) first.

The first publicly available description of HTML was a document called ["HTML Tags"](#), first mentioned on the Internet by Tim Berners-Lee in late 1991.<sup>[6]</sup>

# EXISTING SYSTEM

Mostly the HTML Programming Language will be written in Software Like (TextEditor,VScode) , Its difficult to understand for normal people and many people who face problem like how this software will be work they can't understand how I can run the program who to complie the program .

In this project is simple interface ,its easy to understand how this project will be run .Just you can text the anything and choose the option and CLICK the SUBMIT BUTTON and my project convert into html .

## This is our list of basic HTML tags:

- <a> for link.
- <b> to make bold text. <strong> for bold text with emphasys.
- <body> main HTML part.
- <br> for break.
- <div> it is a division or part of an HTML document.
- <h1> ... for titles.
- <i> to make an italic text.
- <img> for images in document.

## What does it mean to Convert to HTML?

What Does HTML Converter Mean? An HTML converter is a software program that, in simple terms, **converts a basic text file into HTML code**. HTML is the universally accepted standard language in which Web pages are written.

# **NEED AND SCOPE OF COMPUTER SYSTEM**

This Project system will provide an easy approach to share a information of html. This project will help the people understand the HTML language in easy way. Basic Tags you can generate in HTML .This project just like social work. With help of this project many people will be interact In coding and HTML everyone can studies simple html command or tags, many can studies basic and can be create website and can be real hosted website. This will understand how tags and command will be work .

The TEXT TO HTML project is to provide better facility to the students and teachers to bring out the easy to learn HTML Language within home ,college and school. It will reduce the time to learn the html.

## **What is use HTML?**

HTML is **the language for describing the structure of Web pages**. HTML gives authors the means to: Publish online documents with headings, text, tables, lists, photos, etc. Retrieve online information via hypertext links, at the click of a button.

### **1. HTML is Easy to Learn and Use**

HTML is very easy to learn and understand. HTML is the first and foremost language that the person will go through for the one who is learning web development. It has simple tags, and there is no hectic of case sensitivity in HTML. It simply has some tags that serve a specific purpose, and that's it. One can easily understand other's code and can

make changes in it if required as there is not a lot more to understand in it. Moreover, it does not throw any error or create any problem like other programming languages if the developer forgets to close the tags or make some mistakes in code.

## **2. HTML is Free**

One of the biggest advantages of HTML is that it is free of cost, and there is no need to purchase specific software. One should not have to deal with different plugins required to work on any software as HTML does not require any plugins. So it is very cost-effective from a per business perspective as there is no cost of purchasing the license if the whole website is developed in HTML language.

## **3. HTML is supported by all Browsers**

HTML supports almost all browsers around the globe. So there is no need to worry about the website written in HTML for the browser support as the website would easily show up in all the browsers if the program keeps in mind to optimize the website for the different browsers. HTML provides an easy way to optimize the website in HTML according to browsers to the web developers.



# **PROPOSED SYSTEM**

- **OBJECTIVES**
- **REQUIREMENT ENGINEERING**
- **REQUIREMENT GATHERING**
- **SRS**

## **OBJECTIVE**

The objective of this project is to create the interest in common people of India. As it is modern technology and it should be welcoming to its new potential Student or Common people.

This will create enough interest of student or common people because it is welcoming and the purpose of some coders gets information, especially with thousands of initial company offering each year. All programmer rely on opensource blockchain technology to operate in a decentralized format. Blockchain technology makes it easier to authorize write programmer between individuals without requiring a third party.

## **HTML Learning Objectives**

### **Learning Objectives**

**If you do well in this unit, you should be able to:**

- **Insert a graphic within a web page.**
- **Create a link within a web page.**
- **Create a table within a web page.**
- **Insert heading levels within a web page.**
- **Insert ordered and unordered lists within a web page.**
- **Use cascading style sheets.**
- **Create a web page.**
- **Validate a web page.**
- **Publish a web page.**

# **REQUIREMENT ENGINEERING**

HTML is a powerful coding tool for Web development. It is used along with CSS to design and build websites. So, it goes without saying that if you wish to make it big in the domain of Web development, you must get your base right – learn HTML. Thankfully, HTML has one of the simplest learning curves, and you don't even need any prior programming experience to learn HTML!

Although it may seem daunting in the beginning, remember to progress by taking baby steps. The best way to learn a new language or a new skill is to practice as you learn. This holds particularly true for programming. Thus, it is an excellent idea to build HTML projects to strengthen your professional portfolio.

## **These are the requirement engineering tasks**



**Requirements Engineering Tasks: The software requirements engineering process includes the following steps of activities:**

- **Inception.**
- **Elicitation.**
- **Elaboration.**
- **Negotiation.**

- **Specification.**
- **Validation.**
- **Requirements Management.**

# **REQUIREMENT GATHERING**

Requirements gathering is **the process of determining what your projects need to achieve and what needs to be created to make that happen**. You're probably familiar with the fact that everybody has their own common project assumptions about what a project should include.

## **The 6-step requirements gathering process**

- 1. Step 1: Assign roles. The first step in requirements gathering is to assign roles in your project. ...**
- 2. Step 2: Meet with stakeholders. ...**
- 3. Step 3: Gather and document. ...**
- 4. Step 4: List assumptions and requirements. ...**
- 5. Step 5: Get approval. ...**
- 6. Step 6: Monitor progress.**

## **What is requirements gathering in project management?**

Requirements gathering is the process of identifying your project's exact requirements from start to finish. This process occurs during the project initiation phase but you'll continue to manage your project requirements throughout the entire project timeline.

Requirements gathering typically happens during the project brief or initial kick-off meeting.

Some questions include:

- **How long will our project schedule be?**

- **Who will be involved in the project?**
- **What risks may we face in this project?**

Requirements gathering shouldn't be complex, but it's an important component of the project initiation process.

# **SRS**

Visual code is a streamlined code editor with support for development operations like debugging, task running and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs

Scalable Vector Graphics(SVG) is a web-friendly vector file format. As opposed to pixel-based raster files like JPEGs, vector files store images via mathematical formulas based on point and lines on grid. It not just their resizing abilities that make SVGs hugely popular with web designer

## **What is SRS?**

A software requirements specification (**SRS**) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

## **Why SRS?**

In order to fully understand one's project, it is very important that they come up with an **SRS** listing out their requirements, how are they going to meet them and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

Main Elements. Depending on the methodology employed (agile vs waterfall) the level of formality and detail in the **SRS** will vary, but in general an **SRS** should include a description of the functional

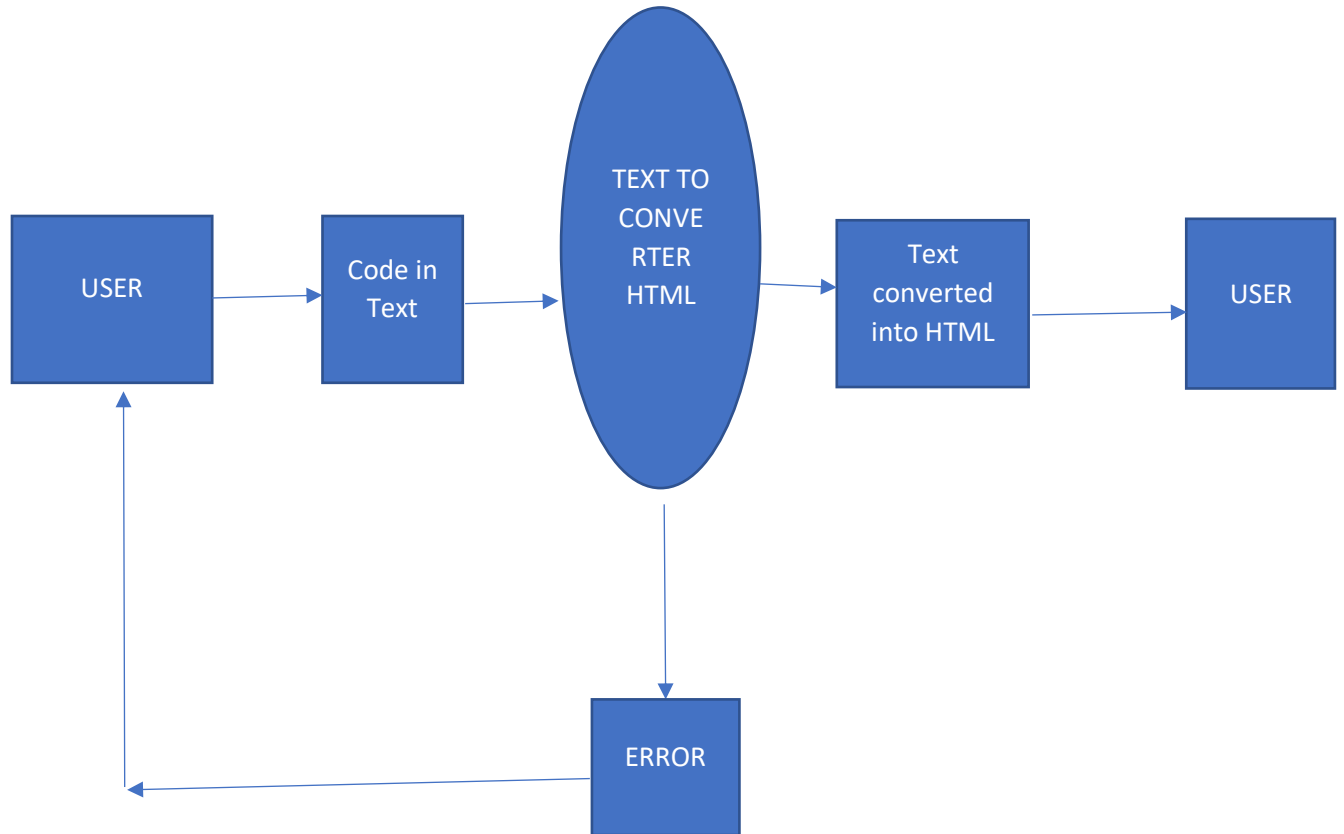
requirements, system requirements, technical requirements, constraints, assumptions and acceptance criteria.



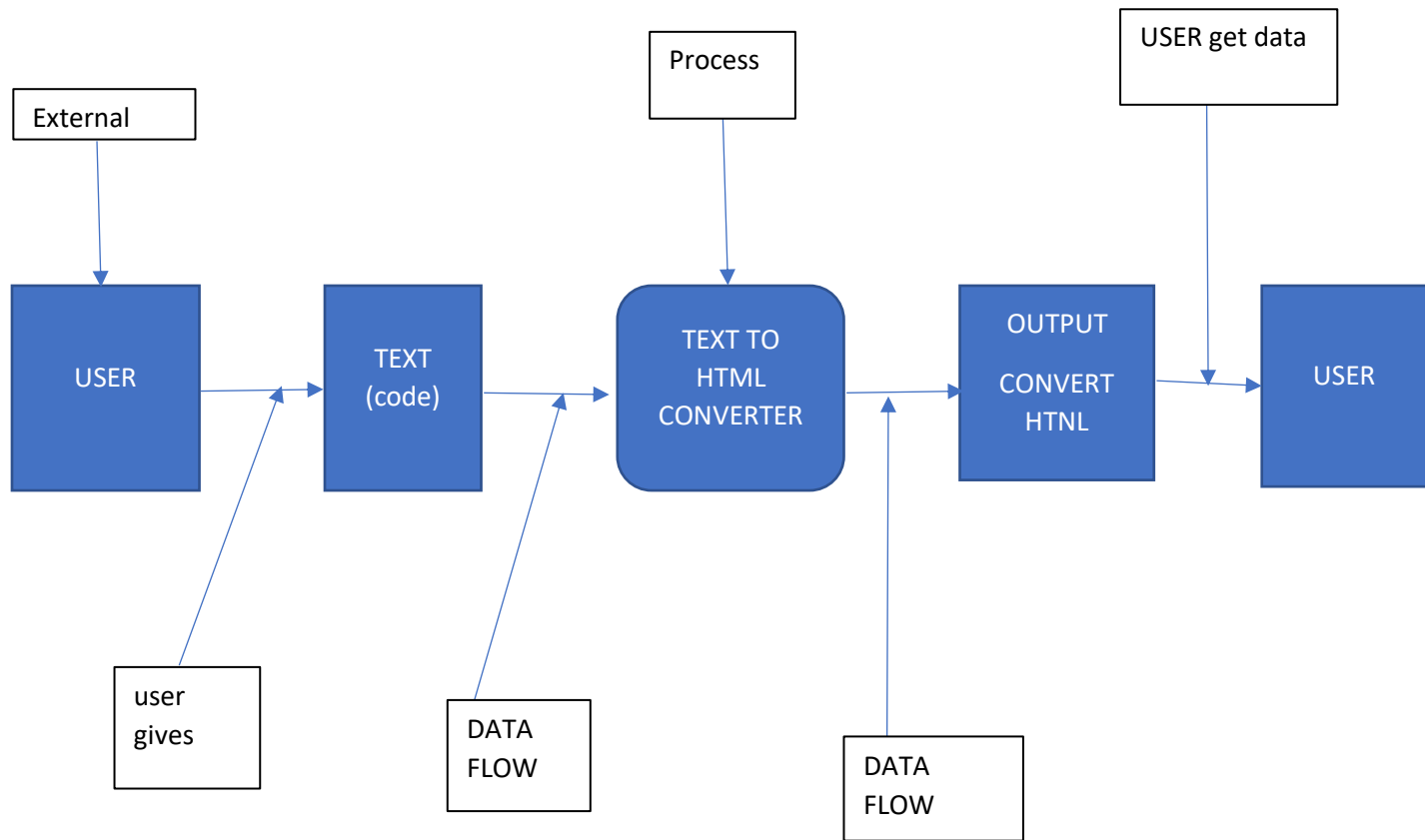
# **SYSEM ANALYSIS**

- **SYTEM DIAGRAM**
- **DFD**
- **UML**

# SYSTEM DIAGRAM



# DFD



**ERD**

**UML**

# **SYSTEM DESIGN**

- **INPUT DESIGN**

# INPUT DESIGN

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>TexttoHtml</title>


<!-- ++++++ bootstrap cdn ++++++ -->

<link                                     rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxt2MZw1T"
crossorigin="anonymous">


</head>

<style>

  #cke_id_body{

    width:inherit!important;

  }

  .htmleditor p{

    font-weight: 900;

    font-size: 20px;

  }

  .texteditor p label{

    font-weight: 800;
```

```
        font-size: 18px;
    }
    #htmldata{
        font-weight: 600;
    }
</style>
<body>
```



```

<!-- -----Navbar start ----- -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a class="navbar-brand" href="#">TexttoHtml</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
    data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent"    aria-expanded="false"    aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home </a>
            </li>
            <li class="nav-item active">
                <a class="nav-link" href="#">About </a>
            </li>
            <li class="nav-item active">
                <a class="nav-link" href="#">Contact</a>
            </li>
        </ul>
    </div>
</nav>

```

```

<!-- -----Navbar Ends----- -->

```

```

<div class="container-fluid">
    <div class="row">

```

```
<div class="col-lg-6 col-md-6 col-sm-6 col-6">
    <div class="texteditor">
        {{form.media}}
        {{form.as_p}}
    </div>
    <input type="submit" class="btn btn-info" onclick="TextConvert()">
</div>
<div class="col-lg-6 col-md-6 col-sm-6 col-6">
    <div class="htmleditor">
        <p>HTML Editor</p>
        <div class="form-group">
            <textarea class="form-control " rows="16" id='htmldata'>
            </textarea>
        </div>
    </div>
</div>
</div>
</div>

</body>
</html>
```

```
<script>

    function TextConvert(){

        var x = CKEDITOR.instances['id_body'].getData();

        var y=document.getElementById('htmldata');

        y.innerHTML=x;

    }

</script>

<!-- ++++++ bootstrap cdn ++++++ -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"

    integrity="sha384-

q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"

    crossorigin="anonymous"></script>

<script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"

integrity="sha384-

U02eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"

crossorigin="anonymous"></script>

<script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"

integrity="sha384-

JjSmVgyd0p3pXB1rRibZUAYoIIy60RQ6VrjIEaEf/nJGzIxFDs4x0xIM+B07jRM"

crossorigin="anonymous"></script>
```

# **IMPLEMENTATION**

- **SYSTEM REQUIREMENT**
- **HARDWARE**
- **SOFTWARE**
- **USER GUIDELINE**

# **SYSTEM REQUIREMENT**

A broadband internet connection with at least 2 mbps upstream bandwidth for uploading large files. Check your upload speed on speedtest.net. If your upload speed is less than 2 Mbps, it may be difficult or impossible for you to upload large files to our website. Contact your internet service provider to inquire about other plans they may offer that have more upstream bandwidth.

An up-to-date web browser. We support recent versions of Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Microsoft Internet Explorer, and iOS and Android mobile browsers.

An up-to-date operating system. We support Windows 7 or newer and Mac OS 10.8 or newer. We recommend that you keep your computer up-to-date as far as operating system security patches and anti-virus software are concerned.

# **HARDWARE REQUIREMENT**

- Computer = 4 GHz minimum,multi-core processor.
- Memory(RAM) = At least 4GB, preferably higher, and commensurate with concurrent usage.
- Hard disk space = At least 10GB.
- Screen resolution = 1280x1024 or larger
- Application window size = 1024x680 or larger
- Processor = Intel Pentium 4 or later.

# **SOFTWARE**

Visual Studio code = A standalone source code editor that runs on Windows, macOS and Linux. The top pick for javascript and web developers, with extensions to support just about any programming language.

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications.

Scikit-learn is probably the most useful library for machine learning in python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction.

Google Chrome is a cross-platform web browser developed by Google. Most of chrome's source code come from Google's free and open-source software project chromium but chrome is licensed as proprietary freeware. Web kit was the original rendering engine but Google eventually forked it to create the Blink engine.

# **USER GUIDELINE**

While the appearance of your website is certainly important, most people aren't coming to your site to evaluate how slick the design is. They want complete some action or to find some specific piece of information.

Keep the structure of your primary navigation simple and near the top of your page . Include navigation in the footer of your site. In addition to keeping your navigation similar across all of your sites pages.

Backgrounds, color schemes, typerfaces, and even the tone of your writing are all areas where consistency has a positive impact on usability and UX.

Site has to be compatible with the many different devices that your visitors are using. In the tech world this is known as responsive design. Responsive design means investing in a highly flexible website structure. On a responsive site content is automatically resized and reshuffled to fit the dimension of whichever device a visitor happen to be using. This can be accomplished with mobile-Friendly HTML templates.



# **OUTPUT**

## **Launching Project in Terminal**

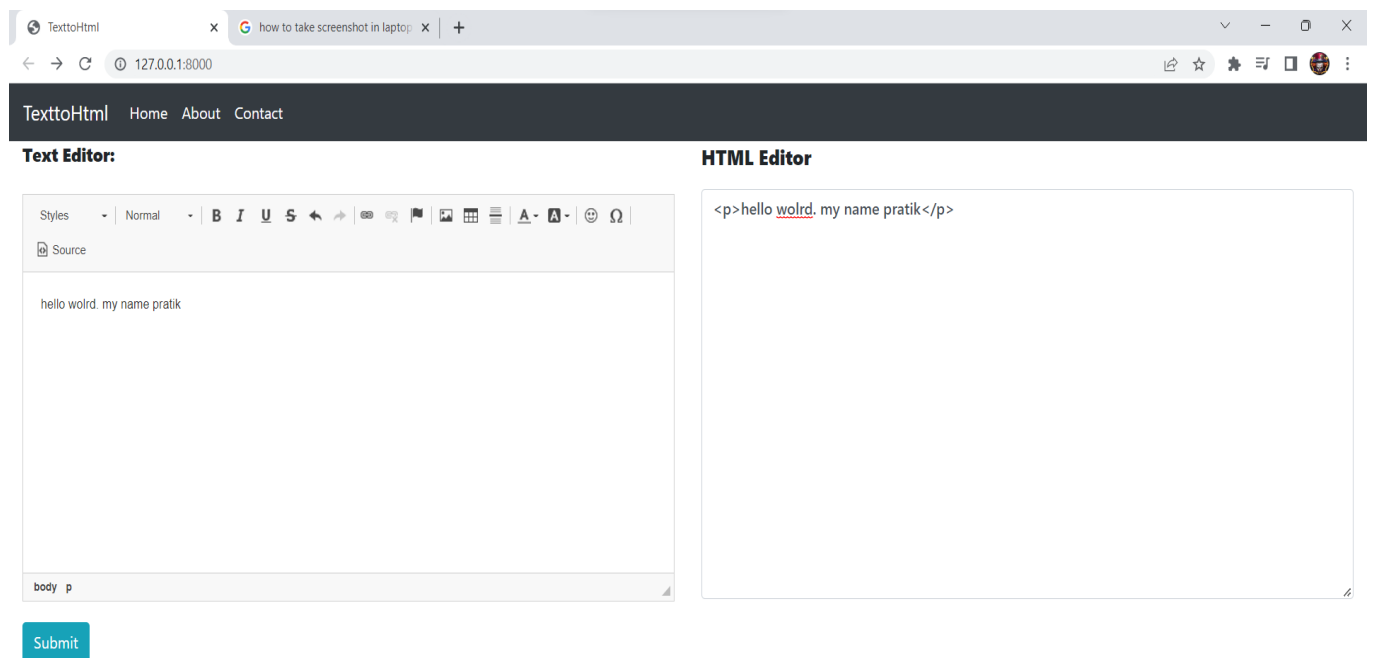
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

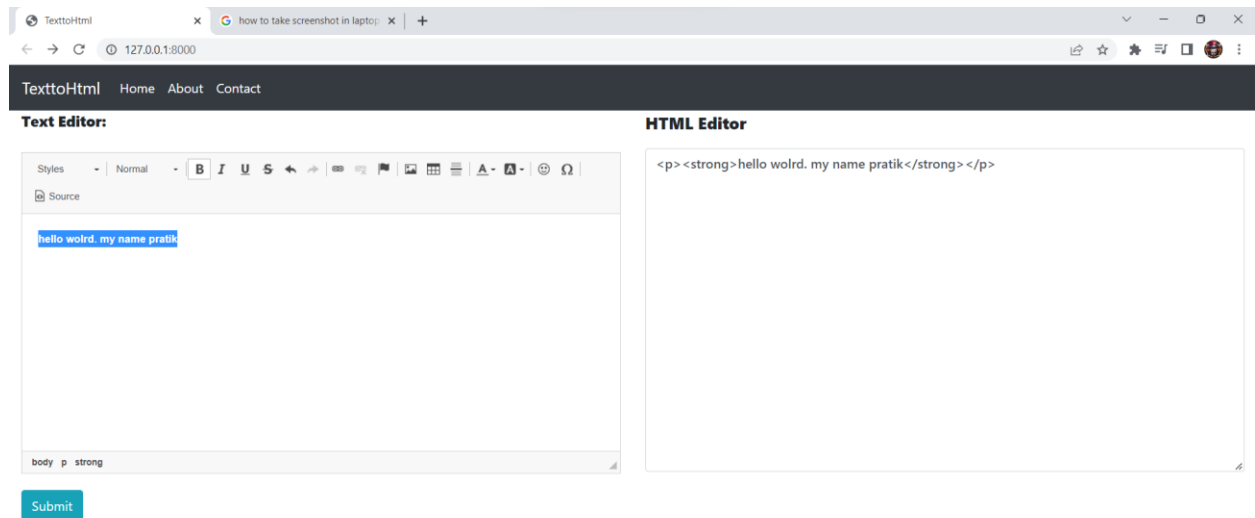
PS P:\project\texttohtml> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

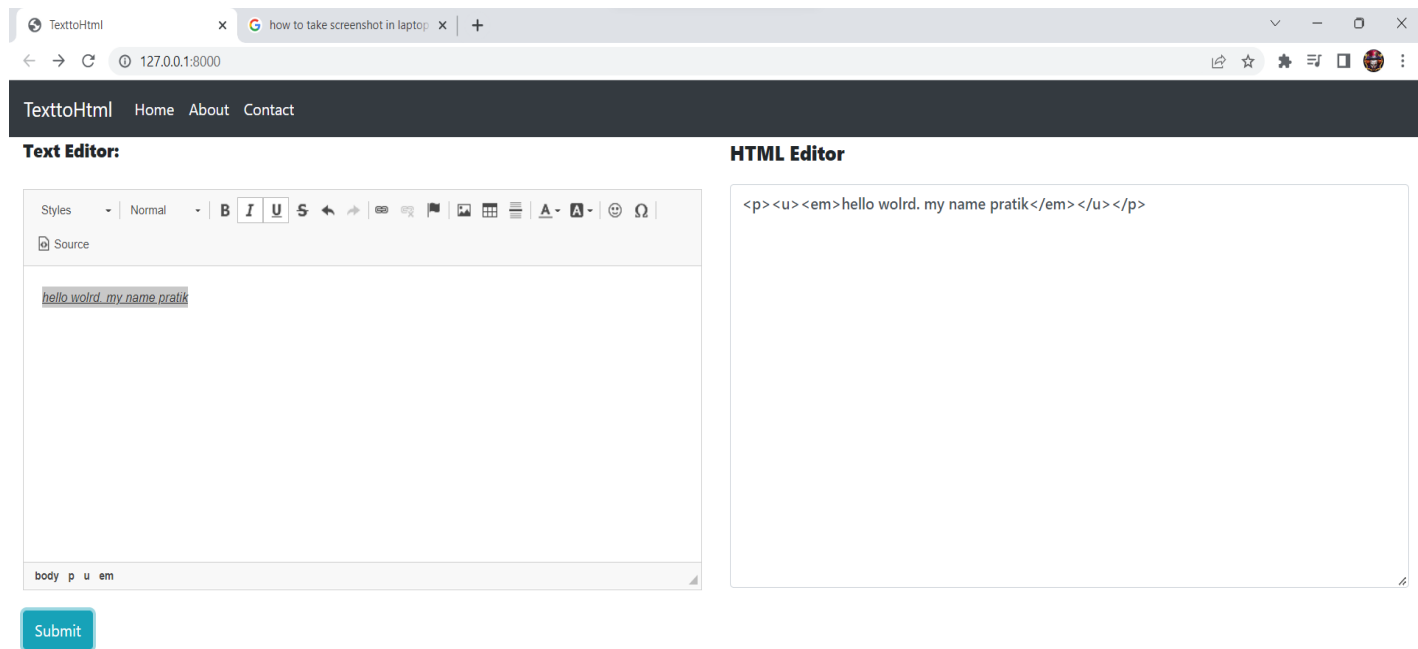
System check identified no issues (0 silenced).
December 01, 2022 - 08:53:07
Django version 4.1.3, using settings 'texttohtml.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[01/Dec/2022 08:53:17] "GET / HTTP/1.1" 200 4864
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor-init.js HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/ckeditor.js HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/config.js?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/skins/moono-lisa/editor.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/lang/en.js?t=M6K9 HTTP/1.1" 304 0
Not Found: /favicon.ico
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/styles.js?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /favicon.ico HTTP/1.1" 404 2227
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/plugins/scayt/skins/moono-lisa/scayt.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/plugins/scayt/dialogs/dialog.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/plugins/tableselection/styles/tableselection.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/plugins/dialog/styles/dialog.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/plugins/copyformatting/styles/copyformatting.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 08:53:17] "GET /static/ckeditor/ckeditor/skins/moono-lisa/icons.png?t=445cf24ebd HTTP/1.1" 304 0
[01/Dec/2022 08:53:18] "GET /static/ckeditor/ckeditor/contents.css?t=M6K9 HTTP/1.1" 304 0
Not Found: /flags
[01/Dec/2022 09:01:54] "GET /flags HTTP/1.1" 404 2209
[01/Dec/2022 09:12:07] "GET /static/ckeditor/ckeditor/skins/moono-lisa/dialog.css?t=M6K9 HTTP/1.1" 304 0
[01/Dec/2022 09:12:07] "GET /static/ckeditor/ckeditor/plugins/table/dialogs/table.js?t=M6K9 HTTP/1.1" 200 8956
[01/Dec/2022 09:12:07] "GET /static/ckeditor/ckeditor/skins/moono-lisa/images/close.png HTTP/1.1" 304 0
[01/Dec/2022 09:13:35] "GET /static/ckeditor/ckeditor/plugins/magicline/images/icon.png?t=M6K9 HTTP/1.1" 200 133
[01/Dec/2022 09:14:50] "GET / HTTP/1.1" 200 4864
[01/Dec/2022 09:17:21] "GET / HTTP/1.1" 200 4864
[01/Dec/2022 09:17:25] "GET /static/ckeditor/ckeditor/plugins/link/dialogs/anchor.js?t=M6K9 HTTP/1.1" 200 2107
[01/Dec/2022 09:17:31] "GET /static/ckeditor/ckeditor/plugins/link/dialogs/link.js?t=M6K9 HTTP/1.1" 200 13234
[01/Dec/2022 09:18:10] "GET / HTTP/1.1" 200 4864
[01/Dec/2022 09:18:28] "GET / HTTP/1.1" 200 4864
```

# Create a Paragraph(Hello world)



# USING A BOLD ON TEXT

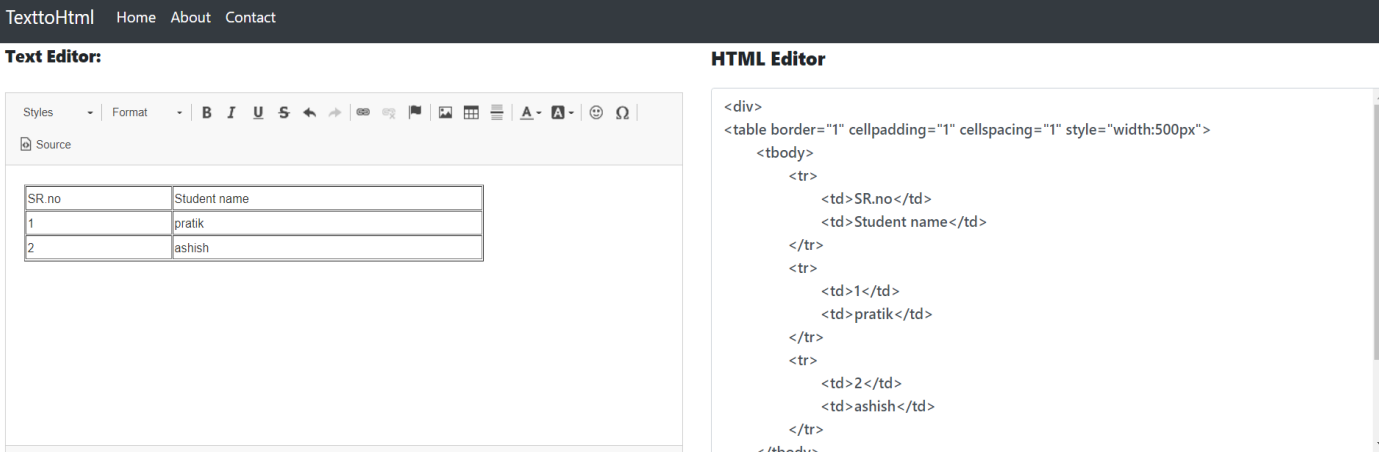




---

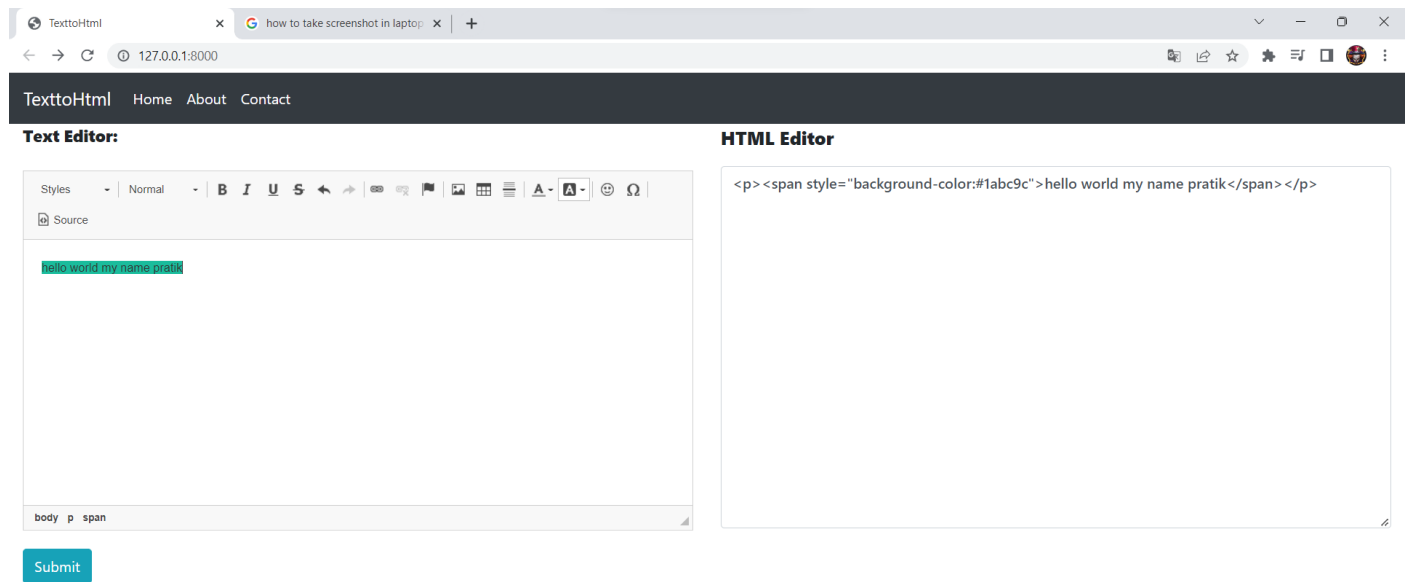
# USING A ITALIC TAG AND UNDERLINE TAG

## CREATING A TABLE IN DIV



The screenshot displays a web browser window with two tabs. The active tab is titled 'TexttoHtml' and shows a URL of '127.0.0.1:8000'. The browser's address bar and navigation buttons are visible at the top. Below the browser window, there are two side-by-side panels. The left panel, titled 'Text Editor:', contains a rich text editor interface. It has a toolbar with various formatting options like bold, italic, underline, and link. Below the toolbar, there is a 'Source' button and a preview of a table. The table has two columns: 'SR.no' and 'Student name'. It contains three rows of data: (1, pratik), (2, ashish), and an empty row at the bottom. The right panel, titled 'HTML Editor', shows the HTML code for the table in the left panel. The code is as follows:

```
<div>
<table border="1" cellpadding="1" cellspacing="1" style="width:500px">
  <tbody>
    <tr>
      <td>SR.no</td>
      <td>Student name</td>
    </tr>
    <tr>
      <td>1</td>
      <td>pratik</td>
    </tr>
    <tr>
      <td>2</td>
      <td>ashish</td>
    </tr>
  </tbody>
</div>
```



---

# SET A BACKGROUND COLOR

TexttoHtml

images - Google Search


127.0.0.1:8000

TexttoHtml Home About Contact

**Text Editor:**

Styles Normal B I U S ↶ ↷ ☰ ☱ ☲ ☳ ☴ ☵ ☶ ☷

Source



body p span

Submit

**HTML Editor**

```
<p><span style="background-color:#1abc9c">hello world my name pratik
</span></p>
```

# ADD IMAGE WITH Paragraph



# CONCLUSION

Websites are playing an important part in improvement of business. 70% of the population is using dynamic and interactive websites because of their eye capturing visual effects. Static websites also have their own place for the platform where there is specific information required. Static websites require less investment as compared to dynamic websites.

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](#).<sup>[9]</sup> 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.

Despite having its own nomenclature, such as naming the callable objects generating the [HTTP](#) responses "views",<sup>[7]</sup> the core Django framework can be seen as an [MVC](#) architecture.<sup>[8]</sup> It consists of an [object-relational mapper](#) (ORM) that mediates between [data models](#) (defined as Python classes) and a [relational database](#) ("Model"), a system for processing HTTP requests with a [web templating system](#) ("View"), and a [regular-expression](#)-based [URL](#) dispatcher ("Controller").

## Components

Also included in the core framework are:

- a lightweight and standalone [web server](#) for development and testing
- a form serialization and validation system that can translate between [HTML](#) forms and values suitable for storage in the database
- a template system that utilizes the concept of [inheritance](#) borrowed from object-oriented programming
- a [caching](#) framework that can use any of several cache methods
- support for [middleware](#) classes that can intervene at various stages of request processing and carry out custom functions
- an internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals
- an [internationalization](#) system, including translations of Django's own components into a variety of languages
- a [serialization](#) system that can produce and read [XML](#) and/or [JSON](#) representations of Django model instances
- a system for extending the capabilities of the template engine
- an interface to Python's built-in [unit test](#) framework

## Bundled applications<sup>[[edit](#)]</sup>

The main Django distribution also bundles a number of applications in its "contrib" package, including:

- an extensible authentication system
- the dynamic administrative interface
- tools for generating [RSS](#) and [Atom](#) syndication feeds
- a "Sites" framework that allows one Django installation to run multiple websites, each with their own content and applications
- tools for generating [Google Sitemaps](#)

- built-in mitigation for [cross-site request forgery](#), [cross-site scripting](#), [SQL injection](#), [password cracking](#) and other typical web attacks, most of them turned on by default<sup>[\[19\]](#)[\[20\]](#)</sup>
- a framework for creating [GIS](#) applications

## **LIMITATION**

### **Django is Monolithic**

Well, this is also a feature for some but for some, it is a drawback. Django has a certain set of files and pre-defined variables. And, you need to learn about those before you [create any project through Django.](#)

Django framework has a certain way to define and perform tasks. It is a logical file structure and easy to learn. But, that also makes it mandatory that you can't use your own file structure. It is because the framework has a way, popularly known as “**The Django way**” of doing things. If you don't follow those rules, you may not be able to deploy anything using Django.

Django framework and server looks for information in these files and will not change that.

Therefore, lots of developers prefer flask than Django, but for that framework, you need to know, backend quite well. Also, you need a lot of time and knowledge to make your own structures and server patterns which are not being done in Django. Here, you only need to spend more time on your unique project rather than other things.

We should use Django in web development because *Django is made for rapid-development*. It is indeed one of the best features of Django, that backs up by the power it provides to the developer without compromising on any functionality and security issues. All that in the simplest and one of the [most powerful languages of the world, Python.](#)

Django framework is the framework of the future which provides integration with most of the current technologies. Also, the community

is constantly developing new and interesting features and functionalities while debugging the project.

So, Django is crowd-tested technology which has been implementing and working on Global-scale applications like Instagram, Pinterest, Disqus, etc.

If you are a professional developer or even a beginner who wants to display improve skills in Python with web-development, then, Django framework is the answer.

This comes to the end of Django advantages and disadvantages tutorial.

## **SUGGESTION**

Generally, we have seen most of the time, there is an input field that provides an autocomplete feature either by typing the specific input data or by seeing a drop-down list from pre-defined options while filling up the form. This feature can be designed & utilized using the HTML `<datalist>` tag in the form. The `<datalist>` tag is used to provide autocomplete feature in the HTML files. It can be used with an input tag so that users can easily fill the data in the forms using select the data. In this article, we will learn to create an input suggestion form using HTML and CSS & will implement the `<datalist>` tag to enable the autocomplete feature in HTML.

## FUTURE ENHANCEMENT

*Some have embraced it, some have discarded it as too far in the future, and some have abandoned a misused friend in favor of an old flame in preparation. Whatever side of the debate you're on, you've most likely heard all the blogging chatter surrounding the "new hotness" that is HTML5. It's everywhere, it's coming, and you want to know everything you can before it's old news. Things like jQuery plugins, formatting techniques, and design trends change very quickly throughout the Web community. And for the most part we've all accepted that some of the things we learn today can be obsolete tomorrow, but that's the nature of our industry. When looking for some stability, we can usually turn to the code itself as it tends to stay unchanged for a long time (relatively speaking). So when something comes along and changes our code, it's a big deal; and there are going to be some growing pains we'll have to work through. Luckily, rumor has it, that we have once less change to worry about. In this article, I'm hoping to give you some tips and insight into HTML5 to help ease the inevitable pain that comes with transitioning to a slightly different syntax. Welcome to HTML5.*

# **BIBLIOGRAPHY**

**GOOGLE.COM**

**YAHOO.COM**

**GETBOOTSRAP.COM**

**YOUTUBE.COM**

**PYTHON.COM**