WEB DEVELOPMENT

By - RADHA S MOHATA

(TCS Intern)

Table of Contents

[Introduction 4](#_Toc70290801)

[Technologies Used 5](#_Toc70290802)

[HTML 5](#_Toc70290803)

[CSS 6](#_Toc70290804)

[Bootstrap 7](#_Toc70290805)

[JavaScript 7](#_Toc70290806)

[jQuery 7](#_Toc70290807)

[JSON 8](#_Toc70290808)

[XML 9](#_Toc70290809)

[REST API 9](#_Toc70290810)

[Built-in python data structures 10](#_Toc70290811)

[SQL 11](#_Toc70290812)

[SDLC 15](#_Toc70290813)

[Waterfall Development Methodology: 17](#_Toc70290814)

[Agile Development Methodology: 18](#_Toc70290815)

[SCRUM 20](#_Toc70290816)

[IT SERVICE MANAGEMNET AND ITIL 22](#_Toc70290817)

[What is an IT Service? 22](#_Toc70290818)

[What is Value? 22](#_Toc70290819)

[What is Warranty? 22](#_Toc70290820)

[What is a service request? 23](#_Toc70290821)

[IT Service Management (ITSM) 23](#_Toc70290822)

[ITSM Processes 23](#_Toc70290823)

[ServiceNow: 24](#_Toc70290824)

[IT Infrastructure Library (ITIL): 25](#_Toc70290825)

[Assignments’ Briefs 25](#_Toc70290826)

[Twitter homepage 25](#_Toc70290827)

[User Registration and storing data in browser’s local storage. 26](#_Toc70290828)

[Fetching data from json file and displaying it using pandas 27](#_Toc70290829)

[Fetching data from xml file and displaying it using pandas: 29](#_Toc70290830)

[Flask API for Post and Get Requests 30](#_Toc70290831)

[Learnings 32](#_Toc70290832)

[JavaScript- 32](#_Toc70290833)

[Pandas- 32](#_Toc70290834)

[Flask- 33](#_Toc70290835)

# Table of Figures

[Figure 1: HTML vs HTML5 8](#_Toc70293408)

[Figure 2: Data transfer from server to client 11](#_Toc70293409)

[Figure 3: Data transfer from client to server 11](#_Toc70293410)

[Figure 4: Working of REST API 12](#_Toc70293411)

[Figure 5: SDLC Life Cycle 19](#_Toc70293412)

[Figure 6: Waterfall Development Methodology 21](#_Toc70293413)

[Figure 7: Agile Development Methodology 22](#_Toc70293414)

[Figure 8: SCRUM 25](#_Toc70293415)

[Figure 9: Twitter Home Page 29](#_Toc70293416)

[Figure 10: Registration Form 30](#_Toc70293417)

[Figure 11: Form data stored in the browser’s local storage 30](#_Toc70293418)

[Figure 12: Residents of a country json data 31](#_Toc70293419)

[Figure 13: Representation of json data in tabular form 32](#_Toc70293420)

[Figure 14: Employee data xml file 33](#_Toc70293421)

[Figure 15: Tabular representation of xml data. 33](#_Toc70293422)

[Figure 16: Flask server 34](#_Toc70293423)

[Figure 17: Response sent by the server. 35](#_Toc70293424)

# Introduction

The web development process includes web design, web content development, client- side/server-side scripting and other related tasks.

The web designing part which includes the frontend technologies like HTML, CSS, javascript, bootstrap. The structure is defined by the HTML.

The style and responsiveness to this structure is given using CSS and bootstrap.

Javascript and jQuery are used to provide functionality to the components. Once the frontend is ready the backend needs to be setup.

In the backend the major part is connecting the database. There are various sql and no sql databases that are being used by the servers for storing the information.

As the backend development is completed, the whole site is now ready for the users to use. Before the users can use the site, the site must be deployed on the server.

On deployment the client and server both can interact. This interaction between client and server is carried out through various HTTP requests. The data is transported in json/xml formats through the REST APIs.

The site developed must meet certain standards. Deployment of the site is not the end task in web development. One needs to maintain the site and update it as per the requirements.

# Technologies Used

## HTML

HTML stands for *Hyper Text Markup Language*.

It is the standard markup language for creating Web pages.

It describes the structure of a Web page. it consists of a series of elements that tell the browser how to display the content.

These elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

 HTML has many updates over time, and the latest [HTML](https://www.javatpoint.com/html-tutorial) version is [HTML5](https://www.javatpoint.com/html5-tutorial).

Difference between HTML and HTML5:

|  |  |
| --- | --- |
| **HTML** | **HTML5** |
| It didn’t support audio and video without the use of flash player support. | It supports audio and video controls with the use of <audio> and <video> tags. |
| It uses cookies to store temporary data. | It uses SQL databases and application cache to store offline data. |
| Does not allow JavaScript to run in browser. | Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5. |
| Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc. | Vector graphics is additionally an integral a part of HTML5 like SVG and canvas. |
| It does not allow drag and drop effects. | It allows drag and drop effects. |
| Not possible to draw shapes like circle, rectangle, triangle etc. | Allows to draw shapes like circle, rectangle, triangle etc. |
| It works with all old browsers. | It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc. |
| Older versions of HTML are less mobile-friendly. | HTML5 language is more mobile-friendly. |
| Doctype declaration is too long and complicated. | Doctype declaration is quite simple and easy i.e., <!DOCTYPE html> |
| Elements like nav, header were not present. | New element for web structure like nav, header, footer etc. |

Here’s the diagrammatic representation of how different the html4(html) and html5 version looks.

|  |  |
| --- | --- |
| HTML | HTML5 |

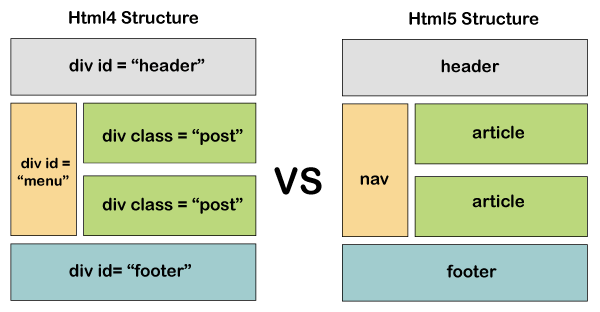


Figure 1: HTML vs HTML5

## CSS

CSS stands for *Cascading Style Sheets*.

It is used to define styles for your web pages, including the design, layout, and variations in display for different devices and screen sizes.

The stylesheets can be included in html in three ways:

Inline: This definition includes describing the element style within its tag using the style command.

E.g.: <p style: “color: #red; font-size:12px;”>

Internal: The internal definition of CSS means the style code is written within the <style> </style> tags which is enclosed in the <head> </head> tags.

External: The definitions are normally saved in external .css files. The external stylesheet is then embedded using the <link> tag. With an external stylesheet file, you can change the look of an entire website by changing just one file!

*The inline CSS definition is given the priority followed by internal and then the external definition.*

## Bootstrap

Bootstrap is a *CSS framework* which provides responsive frontend development.

It contains CSS and javascript based design templates for layouts, buttons, forms, display, navbars, typography and other components.

Use the Bootstrap’s CDN links and embed them in the link and script tags to use bootstrap’s components. If needed, one can override the bootstrap styles.

## JavaScript

JavaScript is a scripting or programming language used to provide interactivity and behavior of web pages.

JavaScript only works inside another application, the web browser.

Enables features like - displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, and many more.

## jQuery

jQuery is a free, highly featured, open source javascript library.

It makes things like *HTML document traversal and manipulation, event handling, animation, and AJAX much simpler with an easy-to-use API* that works across a multitude of browsers.

jQuery can be included using two methods.

1. Use the Google-hosted/ Microsoft-hosted content delivery network (CDN) to include a version of jQuery.
2. Download own version of jQuery from jQuery.com and host it on own server or local filesystem.

*All the jQuery code must be executed only after the page loads completely.*

## JSON

*JavaScript Object Notation* or JSON is a lightweight data interchange format that is independent of any programming language.

Any json file structure is key-value structure. We can access the values with the help of keys.

A sample JSON data looks as follows:

{

“name”: “john”,

“degree”:” B. Tech”,

“subjects”: [“DS”,” OS”,” DBMS”,” TOC”]

}

JSON values can only contain integers, strings, object, null, arrays and Boolean. They cannot be functions.

We need to parse json data before using it. Parsing can be done as-

var obj = JSON.parse(data);

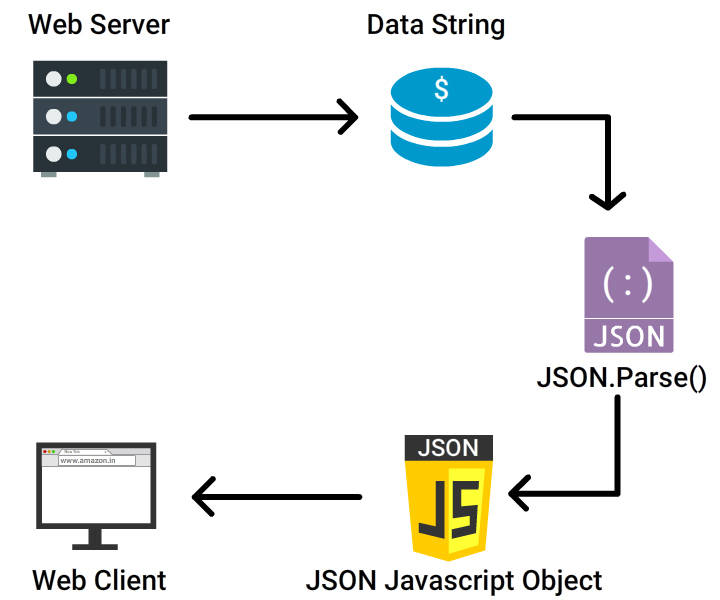


Figure 2: Data transfer from server to client

While receiving data from the web server, the data is always in the string format. On parsing the data with JSON.parse(), the data becomes a javascript object.

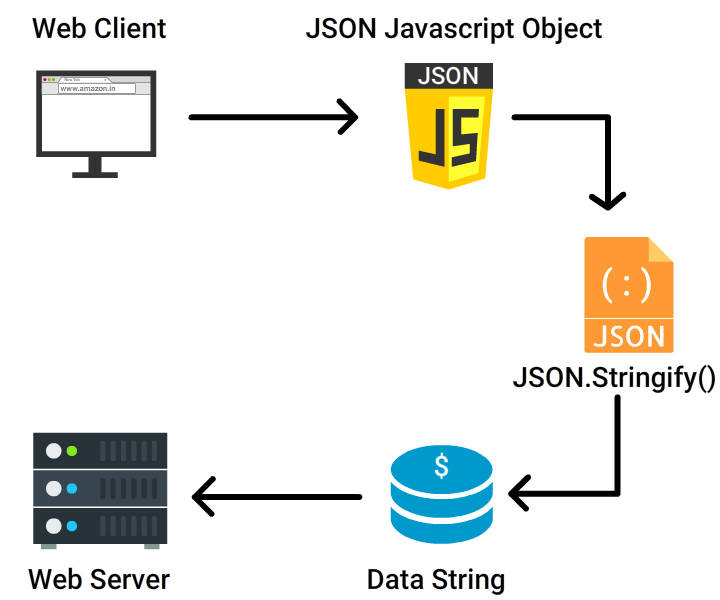


Figure 3: Data transfer from client to server

While sending data back to the web server, the data must be string always. This is done using JSON.stringify().

var myJson = JSON.stringify(obj);

## XML

*Extensible Markup Language* or commonly known as XML is a [markup language](https://en.wikipedia.org/wiki/Markup_language).

It defines a set of rules for storing and transporting data in a human and machine readable [format](https://en.wikipedia.org/wiki/File_format).

The basic building block of an XML document is an element defined by tags.

Being self-descriptive and independent, it simplifies data sharing, data availability and data transporting.

## REST API

REST stands for *Representational State Transfer*. State refers to the state of a resource at a particular time. The REST API is built on the top of HTTP and is thus widely accepted. It uses HTTP requests to access and use the data across the client and server side. It is widely accepted because it uses the HTTP protocols.

REST provides the following advantages:

* Flexibility: Our data in not really tied to the resource or application.
* Scalability: The RESTful servers does not store any client-side information.
* REST is not constrained to return one data format. It can return xml/json.
* Independent of platform and languages.

Below is the graphical representation of how REST API works.

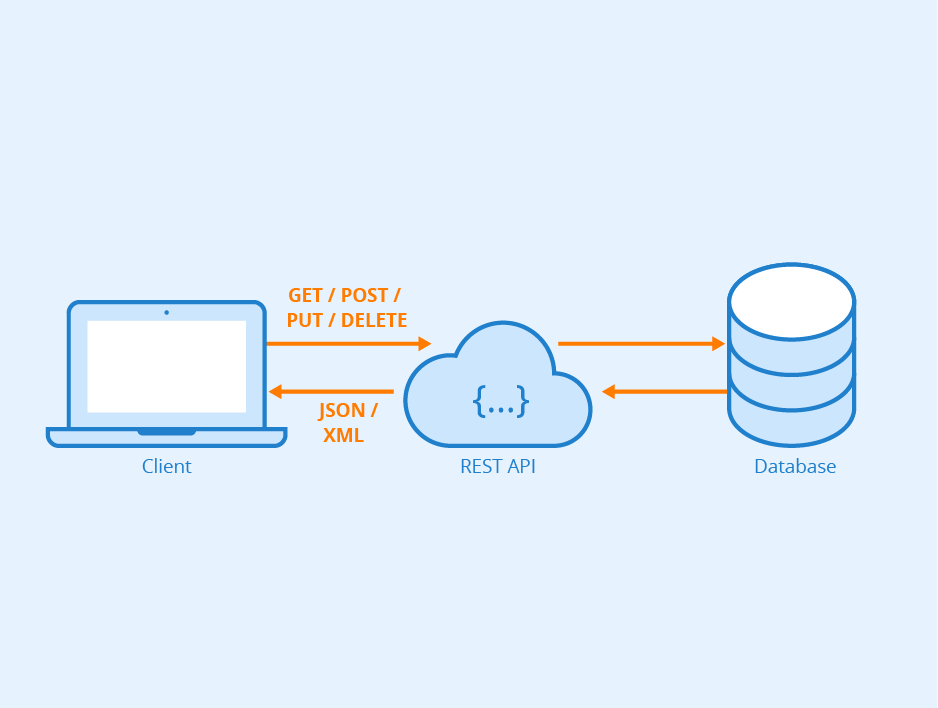


Figure 4: Working of REST API

The client requests the server for the data using the HTTP methods, the server through the REST APIs send the response in the form of JSON/XML data. REST APIs allows communication irrespective of the language on which the client /server is built.

## Built-in python data structures

As the name suggests, these data structures are built-in python which provides various functions for obtaining quicker programming solutions.

There are 4 built-in data structures in python. They are:

1. **List:**

Lists in python are the sequential data structures that can store data of different data types. To create a list, you must use the square brackets and add elements into it. The list elements can be accessed by their index. The index values start from 0.

Python provides various functions to add elements in the list like –

append () - adds all elements pass to it as a single list element.

extend () - adds the elements one-by-one into the list.

insert () - add the element at a given index position.

Other widely used list functions are –

del(index) – to delete the list element at a given index.

Clear () – to empty the entire list.

len () – to get the length of the list.

sorted ()- sort values of the list.

index () – returns the index where the value passed is first encountered.

eg: my\_list = [1, 2, 3, 'example', 3.132, 10, 30]

1. **Dictionary:**

Dictionary is the unique data structure in python that are used to store key-value pairs. They can be created using the curly braces or using the dict () function. Dictionary items can be nested. One need to add data in the form of key-value pairs while working with dictionaries. The keys must be unique and immutable. Dictionary items can be changed, to change the values of the dictionary, you need to use the key.

eg: my\_dict = {'First': 'Python', 'Second': 'Java'}

1. **Set:**

Sets are a collection of unordered elements that are unique. They can be created using curly braces, but here one just must pass single values. One can perform union, intersection, difference, and similar set operations on sets.

eg: my\_set = {1, 2, 3, 4}

1. **Tuple:**

Tuples are same as lists but the data in tuples cannot be changed, i.e., they are immutable. Tuples are created using parenthesis. Accessing elements in tuples is same as it is for accessing values in lists. To append values in tuples, use the ‘+’ operator which will take another tuple to be added to it.

eg: my\_tuple = ("java", "python", "c++")

## SQL

*Structured Query Language* or SQL is the database language that is used to *create, maintain, and retrieve data from the relational databases* like MySQL, Oracle, SQL Server, Post Gre, etc.

CRUD operations i.e., **c**reate, **r**ead, **u**pdate, and **d**elete are the basic functions of storage.

*‘C’* refers to *CREATE* which means to add or insert a new record using the SQL insert statement. SQL uses *INSERT INTO* command to create new records into the table.

Syntax:

|  |  |
| --- | --- |
|  | INSERT INTO <tablename> (column1, column2, ….)  VALUES (value1, value2, ….)  Eg:  INSERT INTO dbo.Demo  VALUES  (1, 'Prashanth’); |

The second letter of CRUD, ‘*R’* refers to *reading* data from the table meaning data retrieval. SQL uses the *SELECT* command to retrieve the data.

Syntax:

SELECT \* FROM <TableName>

Eg:

SELECT \* FROM [Person]. [Address];

*‘U*’ refers to *update* operation. SQL uses the *UPDATE* command to change the existing record of the table.

The basic syntax for an update:

UPDATE <TableName>

SET Column1=Value1, Column2=Value2, …

WHERE <Expression>

*‘D*’ refers to *deleting* or removal of a record from the table. SQL uses the *DELETE* command to perform deletion. To delete related data from the specified table, refer to the below syntax.

DELETE FROM <TableName>

WHERE <Expression>

All the SQL Commands can be broadly classified into five types:

**DDL:**

DDL or *Data Definition Language* consists of SQL commands that can be used to define database schema. It is used to create and modify the structure of database objects in the database.

Eg: CREATE – used to create database or its objects.

DROP – Used to delete objects from the database.

ALTER – Used to alter the structure of database.

TRUNCATE – is used to remove all records from the table including all the space that is allocated.

**DML:**

*Data Manipulation Language* deals with manipulating the data present in the database. Some DML commands are-

INSERT – to insert data into the schema.

UPDATE – to update the existing data within the database.

DELETE – to delete the records from the table.

MERGE – allows you to join a data source table with a target table or view.

**DCL:**

DCL or *Data control Language* includes commands that deal with the rights, permission, and controls of the database system. Some of the data control language commands are:

GRANT – gives user’s access privileges to database.

REVOKE- withdraws user’s access privileges given by using the GRANT command.

**TCL:**

*Transaction Control Language* deals with the transaction within the database. Transaction means a unit of work in SQL.

COMMIT – commits a transaction.

ROLLBACK – if any error occurs, rollbacks a transaction.

**DQL:**

*Data Query Language*, as the name suggests DQL is used for performing query on the data within the database. To obtain some schema relation we use the DQL.

Eg: SELECT: used to retrieve data from the database.

In SQL to perform a certain function there can be more than one command through which you can achieve your result. But these commands are different.

The EXISTS and the IN command differs in the following ways-

**EXISTS VS IN**

|  |  |
| --- | --- |
| Exists | IN |
| EXISTS: This cannot compare the values between the sub-query query and parent query. | IN: It can compare the values between sub-query and parent queries. |
| The output of EXISTS can be either FALSE or TRUE | The output of IN can be TRUE or NULL or FALSE |
| EXISTS is used to determine whether any values are returned or not. | IN can be used as a multiple OR operator. |
| If the sub-query result is large, then EXISTS is faster than IN. | If the sub-query result is less, then IN is faster than EXISTS |
| EXISTS will look for the single positive condition and then will stop the process. | IN compares all the values |

**DELETE VS TRUNCATE**

|  |  |
| --- | --- |
| DELETE | TRUNCATE |
| The DELETE command is used to delete a row in a table. | Used to delete all rows from the table |
| Delete can be used with indexed views. | Truncate cannot be used with indexed views. |
| A data manipulation language command. | A data definition language command. |
| Slower than TRUNCATE | Faster than DELETE |
| Deletes the structure of the row from the table. | Preserves the structure of the table. |

**Views In SQL:**

A view is an efficient way of representing data without the need to maintain it.

It is not an actual table and requires no permanent storage. A virtual table is created and is based on the result-set of an SQL statement.

Views are created using the CREATE VIEW statement.

There are two types of database views:

Dynamic views and Static views.

*Dynamic views* can contain data from one or two tables and automatically include all the columns from the specified table or tables. They are automatically updated.

*Static views* can contain data from multiple tables and the required columns from these tables must be specified in the SELECT and WHERE clauses of the static view. They must be manually updated.

**Materialized Views:**

When the results of a view expression are stored in a database system, they are called materialized views.

Database system uses one of the three ways to keep the materialized view updated:

* Update the materialized view as soon as the relation on which it is defined is updated.
* Update the materialized view every time the view is accessed.
* Update the materialized view periodically.

Materialized view is useful when the view is accessed frequently, as it saves the computation time, as the result are stored in the database beforehand.

Helpful in case where the relation on which view is defined is very large and the resulting relation of the view is very small.

For eg, if we have Employee table and we want to find SUM of employee salary and this result is Stored in Materialized View and used in future.

Materialized view has storage cost and updating overheads associated with it.

## SDLC

*Software Development Life Cycle* is a process or methodology used by software organizations to build the software products.

It gives *the detailed plan about how to design, develop, maintain, replace, and alter or improve a specific software*.

The aim of SDLC is to produce a high-quality product that will meet the customers’ expectations considering all the factors such as time, cost, and quality.

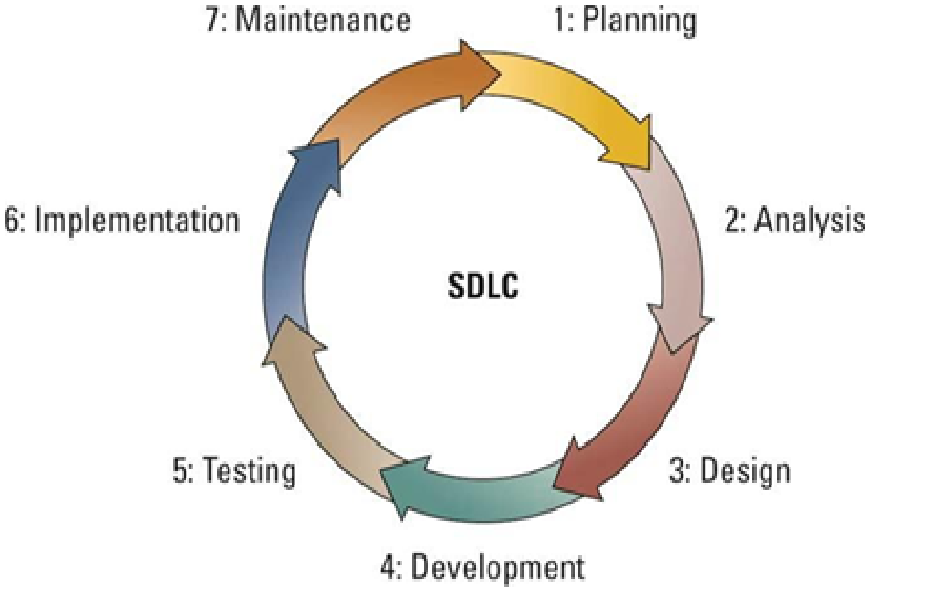
**

Figure 5: SDLC Life Cycle

A typical Software Development Life Cycle consists of the following stages:

1. **Planning Stage:**

Planning comes first. Before starting to build a product, a detailed plan about project, the cost estimations and the requirements is a must to build and deliver the best product. The planning phase involves the entire project team.

1. **Feasibility or Requirement Analysis Stage:**

The second phase involves the customers and the IT specialists. They gather the required data, understand, and then document it. This is done through *Software Requirement Specification* document which consists of all the product requirements to be designed and developed during the project life cycle.

1. **Designing and Prototyping Stage:**

In the third stage, *developers build the blueprint* of how things in the project will work. Design approaches are documented in the Design Document Specification (DDS), and considering all the parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected.

1. **Software Development Stage:**

Once the design and the prototype are selected, the development starts. The program code is generated, databases created, and the software programs are set up. By the end of this stage, you will have a working feature which can be shared with the customers.

1. **Software Testing Stage:**

The code developed by the developers in fourth stage is *tested across various parameter*s like code quality, if the code meets the requirements, evidence of all the guidelines being followed. The products are tested until it reaches the quality standards defined in the SRS.

1. **Implementation and Integration:**

In the sixth stage, the *operation team deploys the code* so that people can use it. In small projects, it will be deploying the code to the web server. Whereas in large projects, it may involve integration of different systems to the main product.

1. **Operation and Maintenance:**

Once the software is implemented and deployed for use, the developers look if there’s any bug or defect in the system. If a bug / defect is found, a bug report is created and then worked upon. Maintaining a system means to ensure its ability to realize its goals.

For a success of this software development process, the organization needs to make sure that all these stages are implemented properly.

SDLC is a general term for the software development. The software organizations take up specific approach to organize and optimize the whole development process and managing the teamwork.

There are various methodologies to improve SDLC, two of these methodologies are *Waterfall Development Methodology and Agile Development Methodology.*

### Waterfall Development Methodology:

This methodology is *strict and linear*. A new stage can only be started if the previous one is completed. While implementing the waterfall model, the exact requirements and needs should be known beforehand.

Below is the graphical representation of the waterfall model:

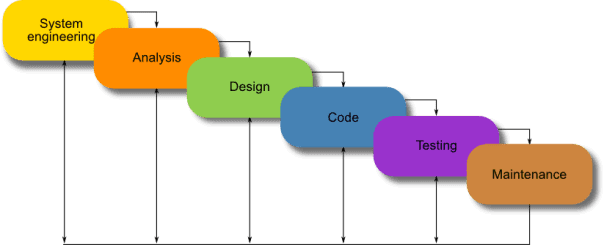


Figure 6: Waterfall Development Methodology

Each phase gradually flows into the next one, the progress is seen as flowing steadily downwards (like a waterfall) through the phases and there is no going back to the previous stage.

The major disadvantages of the Waterfall Model are as follows −

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Difficult to measure progress within stages.

### Agile Development Methodology:

This is the absolute opposite approach to the waterfall model. Agile is a philosophy *to rapidly deploy the software /application in an iterative way*. Owing to the high number of risks and uncertainty in the waterfall model, the agile comes to rescue.

The Agile manifesto states:

*Individuals and interactions* over process and tools

*Working software* over comprehensive documentation

*Customer collaboration* over contract negotiation

*Responding to change* over following the plan.

The Agile works on the above principles and because of which it has several advantages which makes it the most used methodologies of today.

The advantages can be given as:

Persistent software delivery

Inspect and adapt.

Design is important.

Increased stakeholder satisfaction

Welcome to changes at any stage.

Daily interaction.

Below is the graphical representation of the agile model.



Figure 7: Agile Development Methodology

The Agile model adopts Iterative development. Each incremental part is developed over an iteration.

There are different ways to achieve Agile. They are:

* Scrum
* Kanban
* XP (Xtreme Programming)
* FDD (Feature Driven Development).

### SCRUM

Scrum is a framework within which people can address complex problems, while productively and creatively delivering products of the highest possible value.

It is an iteration of plan, build, test and review. The scrum implemented team looks like –

**Product owner**: (think of him as a *manager*) He holds the responsibility to make sure that the application is deployed as and when committed at the same time the application is built exactly as the way it must be built. He is the guy with *idea and the owner of the application* that would be developed.

**Scrum Master:** (think of him as *the team leader*) this person handles the day-to-day operations like running the meetings, handling the tasks that must be done.

**Developers:** Team that does the development task.

Below events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time boxed. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened.

The Scrum Events are:

* [**Sprint**](https://www.scrum.org/resources/what-is-a-sprint-in-scrum)**:**

They are the fixed length of events of one month or less. A new sprint always starts after the pervious one is finished. All the work necessary to achieve the Product Goal, including [Sprint Planning](https://www.scrum.org/resources/what-is-sprint-planning), [Daily Scrums](https://www.scrum.org/resources/what-is-a-daily-scrum), [Sprint Review](https://www.scrum.org/resources/what-is-a-sprint-review), and [Sprint Retrospective](https://www.scrum.org/resources/what-is-a-sprint-retrospective), happen within Sprints. Each sprint may be considered as a short project.

* [**Sprint Planning**](https://www.scrum.org/resources/what-is-sprint-planning)**:**

Includes meeting and discussions. During this the product owner will explain how he imagines the end goal and the application would look like. The [Product Owner](https://www.scrum.org/resources/what-is-a-product-owner) ensures that attendees are prepared to discuss the most important [Product Backlog](https://www.scrum.org/resources/what-is-a-product-backlog) items and how they map to the Product Goal.

* [**Daily Scrum**](https://www.scrum.org/resources/what-is-a-daily-scrum)**:**

The 15 min meeting where you can discuss what happened and where you stand and what you are going to do tomorrow.

* [**Sprint Review**](https://www.scrum.org/resources/what-is-a-sprint-review) **:**

Sprint Review is the *representation of results of the work done* and progress towards the goal. It occurs at the end of user story and the backlog that you have been working on so each one of these user stories are planned of two weeks or four weeks but in majority its of four weeks so that you know exactly what you’re supposed to do for the next two weeks.

* [**Sprint Retrospective**](https://www.scrum.org/resources/what-is-a-sprint-retrospective) **:**

It is *to plan ways to increase the quality and effectiveness*. The team discusses the things that went well during the sprint and problems that occurred and how they were solved.

**Scrum Artifacts**

Scrum’s artifacts *represent work or value*. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

The Scrum Artifacts are:

* [**Product Backlog**](https://www.scrum.org/resources/what-is-a-product-backlog)**:**

Product backlogs consist of *the lists of requirements* that needs to be met to improve product quality. In product backlogs you have the same application iterated over the smaller tasks.

* [**Sprint Backlog**](https://www.scrum.org/resources/what-is-a-sprint-backlog) **:**

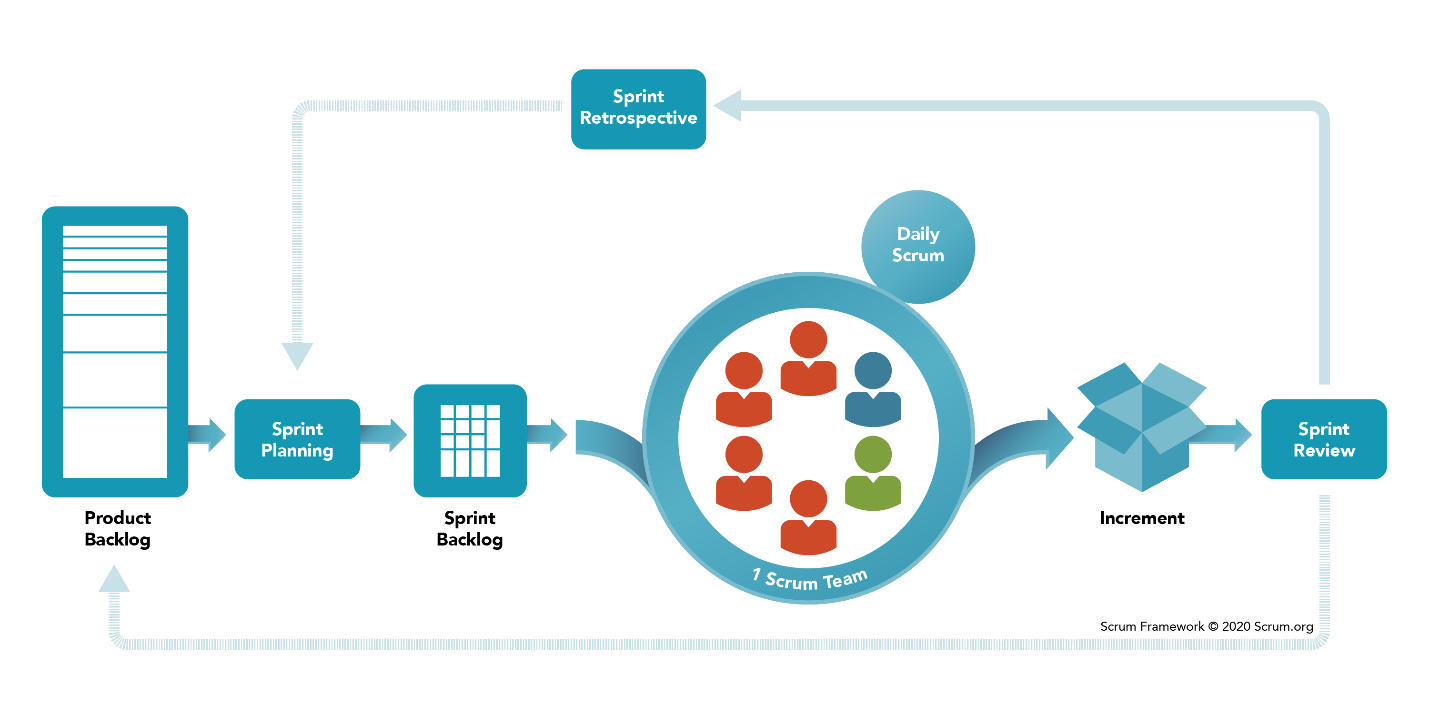
The Sprint Backlog is a plan by and for the developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint to achieve the Sprint Goal.

* [**Increment**](https://www.scrum.org/resources/what-is-an-increment) **:**

Increment is a steppingstone towards the product goal. Each increment can be added to all prior increment which ensures they all work together.

Here’s an example of how Scrum works: John meets with a customer to discuss his company’s needs. Those needs are the product backlog. John chooses the most important tasks to work on in the next two weeks. His team meets in a daily scrum to target work for the day ahead and address roadblocks. At the end of the sprint, John delivers the work, reviews the backlog, and sets the goal for the next sprint. The cycle repeats until the software is complete.

Graphical representation of the scrum -



*Figure 8: SCRUM*

## IT SERVICE MANAGEMNET AND ITIL

### What is an IT Service?

A service is a means of *delivering value to customers* by facilitating outcomes that customers want to achieve without the ownership of specific costs and risks.

There can be different categories of IT services based on the skills (design, build, run) like business process services, application service and infrastructure services.

### What is Value?

Value is generated through exchange of knowledge, information, goods, or services. Values are highly dependent on what effect they have on the costs, outcomes, and risks. While delivering the service, the organization needs to understand what its customer needs. The organization works to deliver the best outcomes.

### What is Warranty?

Warranty is the assurance that a product or service will meet agreed requirements. It answers, ‘how the service performs’ and ‘how fit it is for use’. It mostly deals with the standards of security, availability, capacity, and continuity. Warranty is very essential for the service to reach its desired outcomes.

While providing the services, one must stick to its values and consider the impact of all factors on warranty.

### What is a service request?

A request from a User for information, or advice, or for a Standard Change or for Access to an IT Service. For example, to reset a password, or to provide standard IT Services for a new User. Service Requests are usually handled by a Service Desk, and do not require an RFC to be submitted.

The service request management should follow these guidelines:

* Service requests and their fulfilment should be standardized.
* Policies should be defined for fulfilling service requests.
* The expectations of users regarding fulfilment times and costs should be clearly set, based on what the organization can realistically deliver.
* Opportunities for improvement should be identified and implemented.

### IT Service Management (ITSM)

ITSM is a set of policies, processes, and procedures for managing the implementation, improvement, and support of customer-oriented IT services. It mainly focuses on customer needs and IT services for customers. It describes the processes and the tools that are used to provide the IT services. It helps businesses build the structure around lifecycle of IT services, along with the management and maintenance. It defines roles and responsibilities of every individual involved in the IT services.

Advantages of ITSM:

Improve IT efficiency.

Increase user-satisfaction.

Increased agility

Reduced costs.

More employee satisfaction by clearer roles and responsibilities.

## ITSM Processes

ITSM improves the efficiency of delivering the services in the following ways:

**Incident Management** –

Incident management is a collection of policies, processes, workflows, and documentation that helps IT teams manage an incident from start to finish.

The incident management handles things like prioritizing the incidents and the service requests according to their business impact.

**Problem Management** –

Problem Management processes removes defects from the system and works towards stabilizing the service environment. On the other hand, IT problem management is the process of identifying the root cause leading to one or more incidents and then initiating actions to rectify the issue.

**Change Management** –

Change management is the process of making changes to the IT infrastructure in a standardized and systematic manner. Changes can include replacing or upgrading the capacity of hardware, upgrading to a new version or rolling back to an old version of software, or switching to new vendors of IaaS and PaaS solution. Changes can both be a response to problems and incidents as well as causes of them.

**Release Management** –

Release Management is the *handles software deployments and change initiatives*. It aims at maximizing the values while reducing the negative impact of the changes. Software updates and modules need to release to the production server to make them available for use. The release management makes sure that these releases are done effectively.

### ServiceNow:

**ServiceNow** is a cloud-based software platform for IT Service Management (ITSM) which helps to automate IT Business Management. It is designed based on ITIL guidelines to provide service-orientation for tasks, activities, and processes. It uses machine learning to leverage data and workflows to help businesses become faster and scalable.

It offers the power and flexibility to achieve the goals of the incident and problem management. It provides all the information to the technician to diagnose and repair issues while removing the dependency on spreadsheets and emails.

In software development, an environment refers to a server tier designated to a specific stage in a release process. There are basically 3 environments in software development, namely;

**Development Environment**—

This is where the software is developed. In some situations, this could be the developer’s desktop, in other situations this would be a server shared by several developers working together on the same project. This environment should resemble the production environment as much as possible to prevent issues where the software acts differently on production.

**Test Environment** —

After the application has been developed to an agreed stage it is released to the testing environment. This is where the testers ensure the quality of the application, open bugs, and review bug fixes. This environment must resemble the production environment accurately because this is the last safe place to find and fix environment-related bugs.

**Production Environment**—

This is where the application goes out to the world and become production. Content can be updated from the staging environment into Production Environment, when available, as well as new application functionality and bug fixes release from UAT or staging environment.

### IT Infrastructure Library (ITIL):

ITIL is a framework of best practices for delivering IT services. The systematic approach of ITIL to IT service management help businesses to manage risk, strengthen customer relations, establish cost-effective practices, and build a stable IT environment that allows for growth, scale, and change.

It describes general processes, procedures, tasks, and checklists that can be applied to organizations toward strategy, delivering value and maintaining minimum level of competency. It is used to demonstrate compliance and to measure improvement.

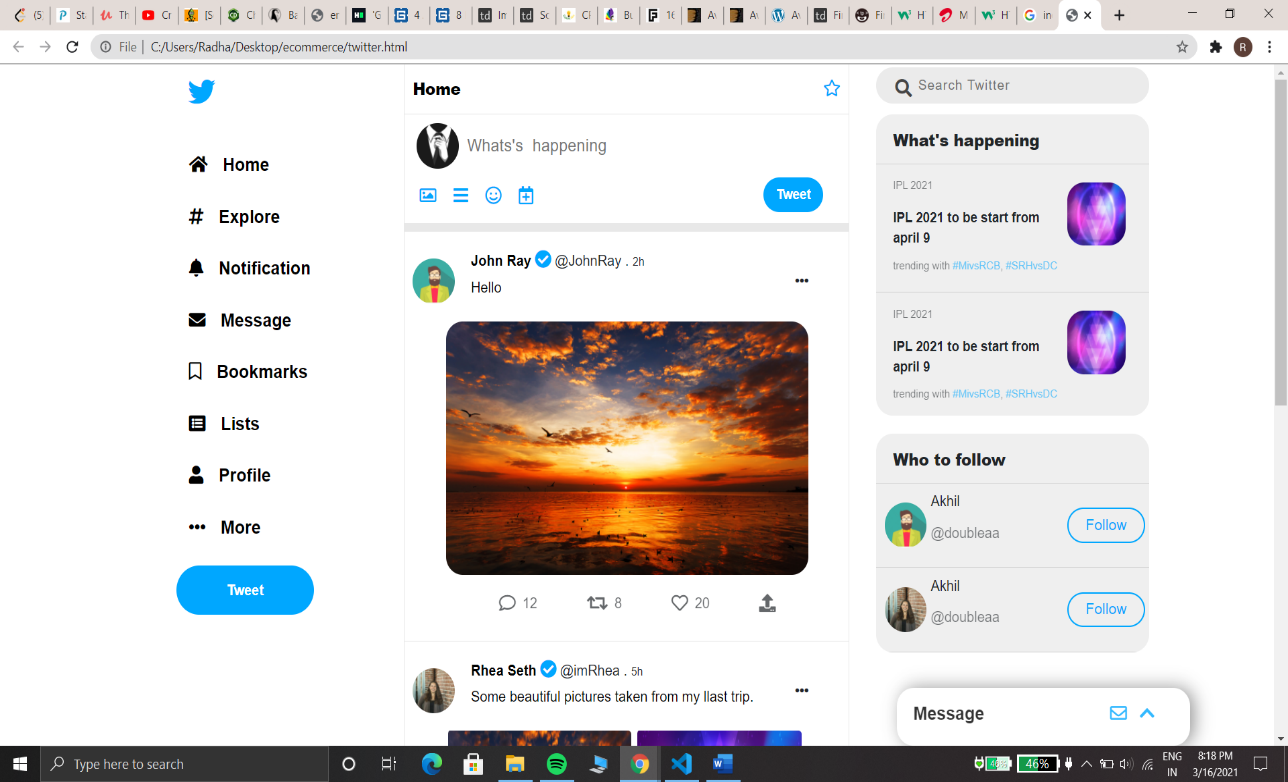
The UK Government’s Central Computer and telecommunication Agency developed this set of recommendations to standardize IT management practices. ITIL, since its first release in 1980, has continuously evolved into many versions and currently has 5 books.

# Assignments’ Briefs

## Twitter homepage:

* Created a responsive twitter homepage using html, CSS, js and jQuery.
* For building a page/site the first thing is understanding and designing the layouts for each part that you need to include in that page/site.
* To make sure your page looks proper on every device, it should be responsive. The page is made responsive using bootstrap or media query.

Here’s the snippet of the twitter homepage:

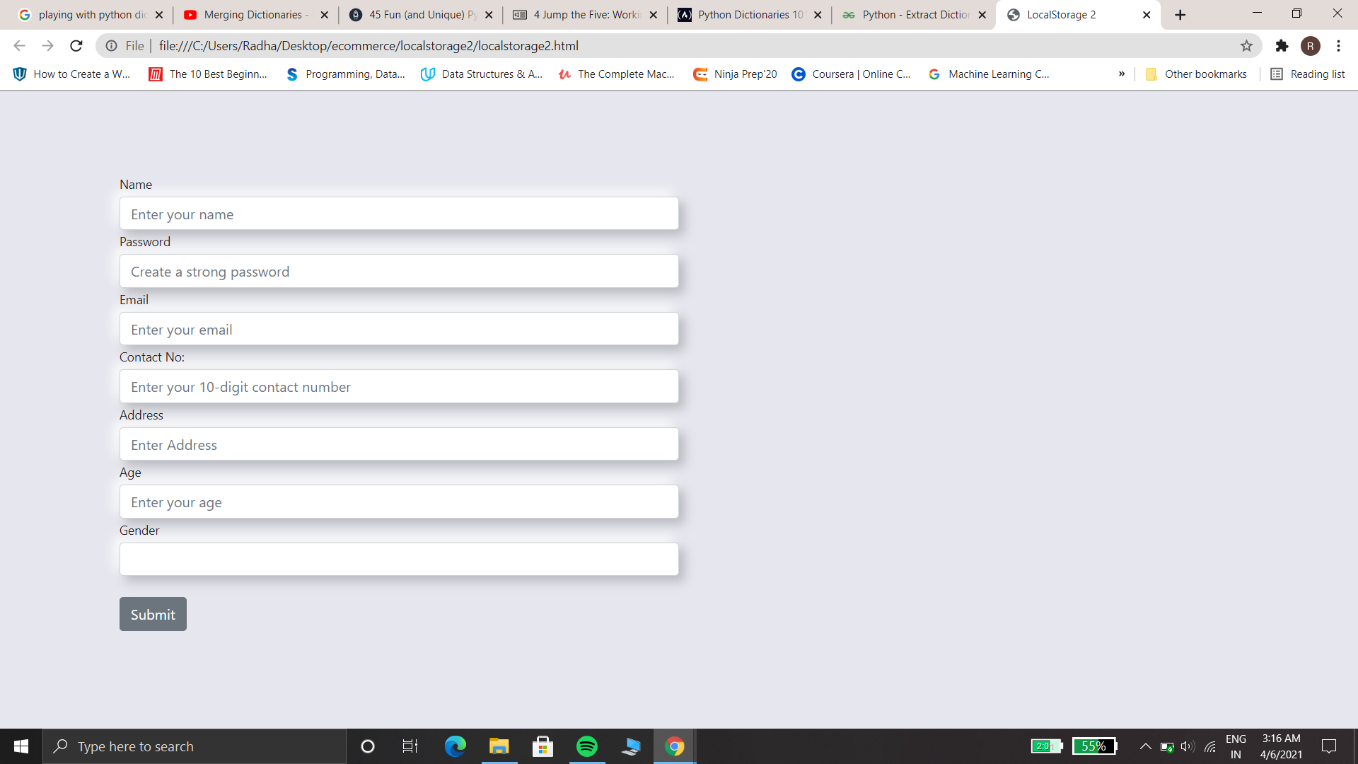


*Figure 9: Twitter Home Page*

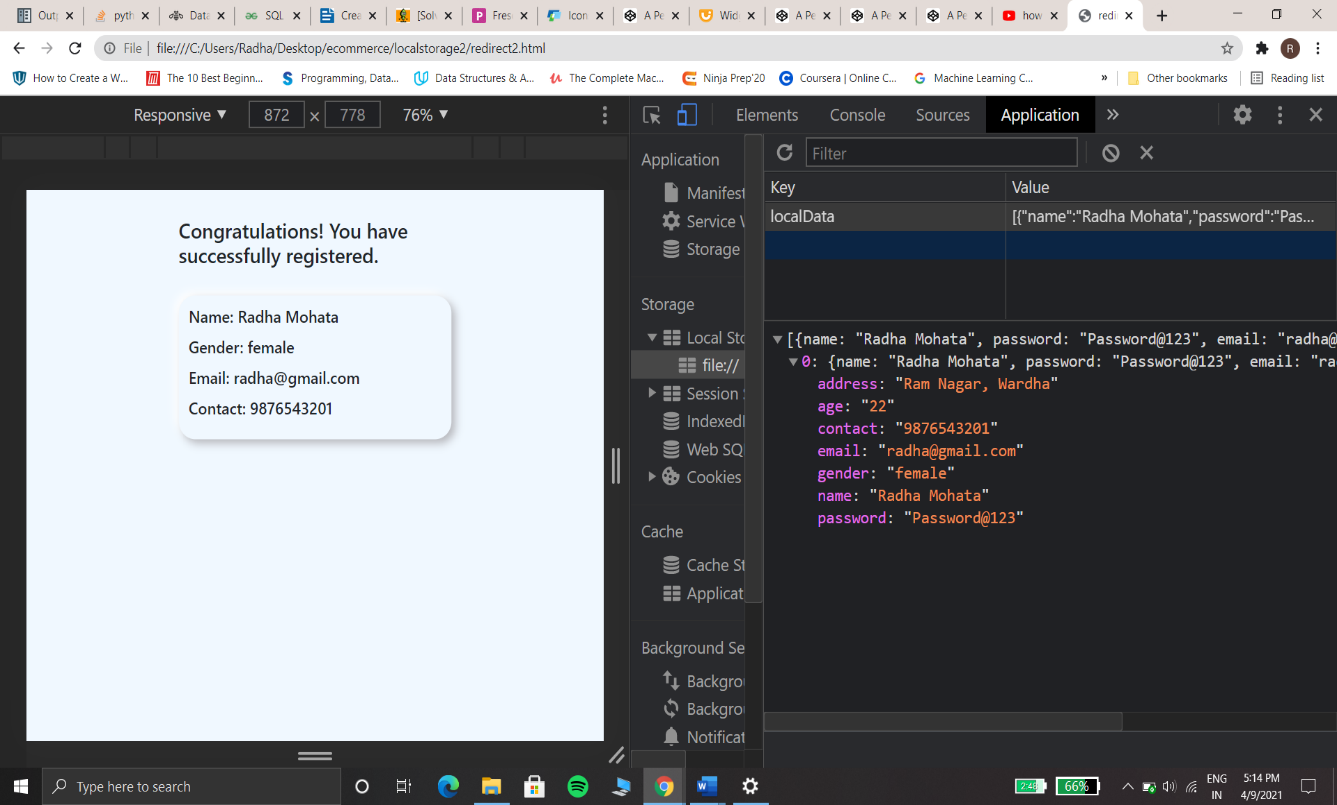
## User Registration and storing data in browser’s local storage:

* The localStorage is a property that allows JavaScript sites and apps to save key/value pairs in a web browser with no expiration date.
* The data stored in the browser will persist even after the browser window is closed.
* The local storage can only save up to 5-10mb of data.
* In Google Chrome, [web storage](https://en.wikipedia.org/wiki/Web_storage) data is saved in an SQLite file in a subfolder in the user’s profile. The subfolder is located at  \AppData\Local\Google\Chrome\User Data\Default\Local Storage on Windows machines.

The data stored in the local storage of the Chrome Browser looks as below:



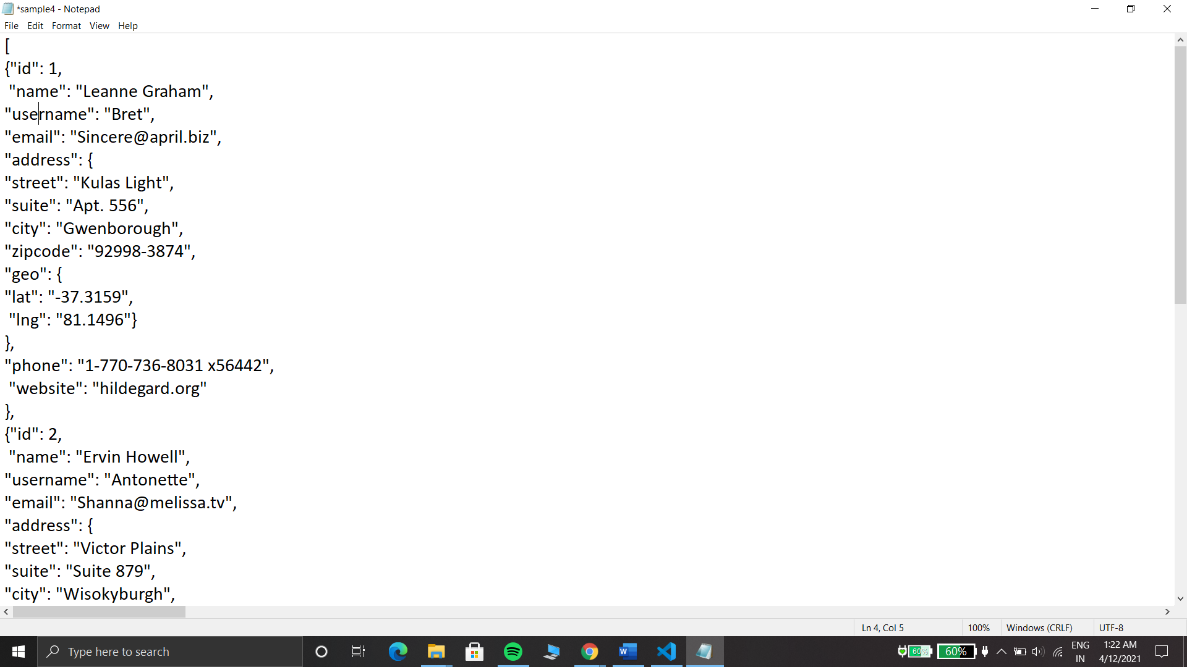
*Figure 10: Registration Form*

*Figure 11: Form data stored in the browser’s local storage*

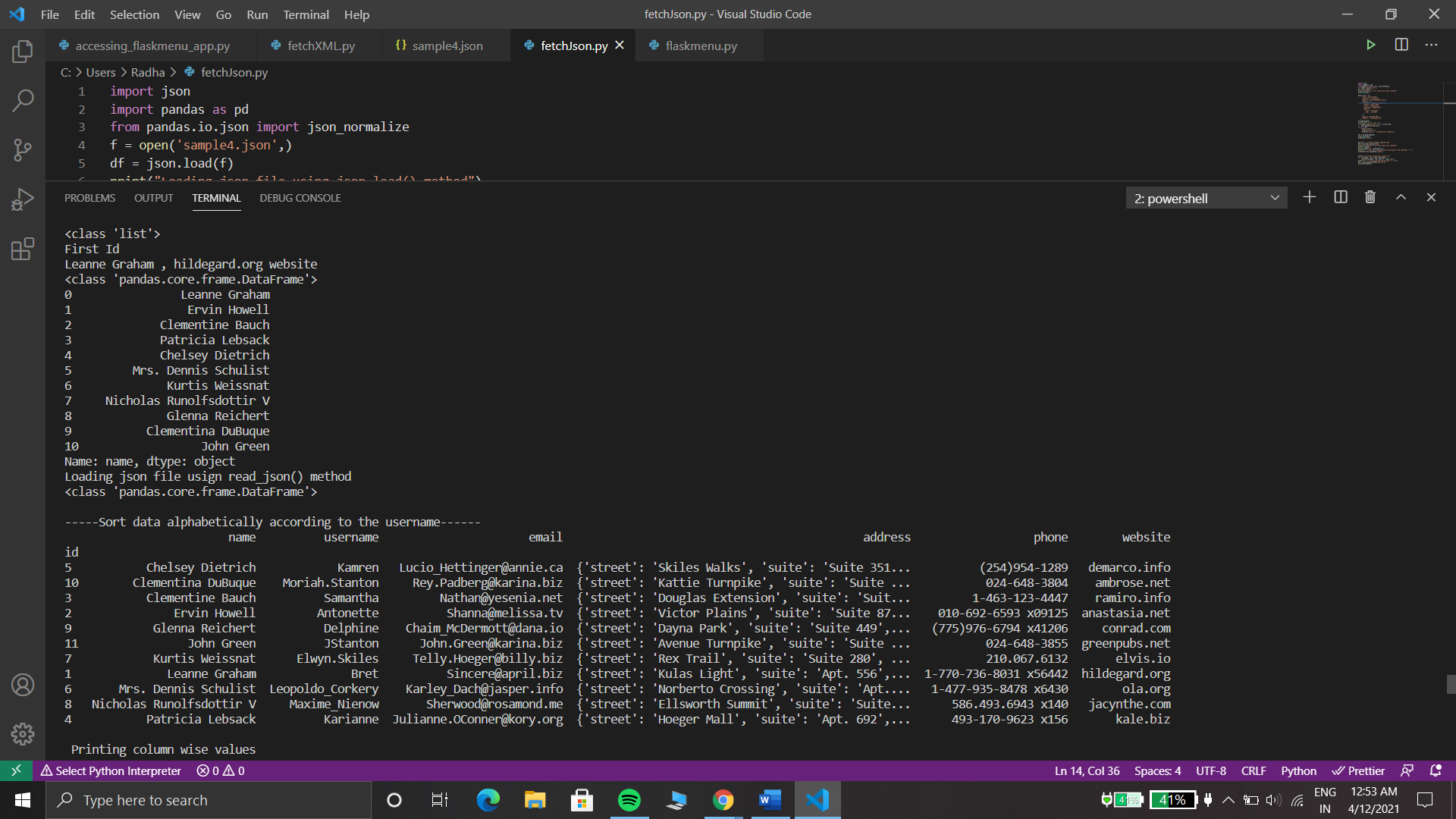
## Fetching data from json file and displaying it using pandas:

* Working with large JSON datasets can be deteriorating, particularly when they are too large to fit into memory.
* In cases like this, a combination of command line tools and Python can make for an efficient way to explore and analyze the data.
* Parsing, writing, and updating data can be easily done using Python’s Data Analysis Library -Pandas. Pandas provide a large set of functions to play with these large data sets. With pandas you can display these data in a tabular format.

A sample json file that contains detailed data of some residents of a country. The address of each resident is represented in a nested structure. The data in Pandas can be read by loading the file and also by using the read json() method. The load() file type is either a list or a dictionary and the read\_json() type is a pandas dataframe. Below snaps shows how one can play with the datasets using pandas.



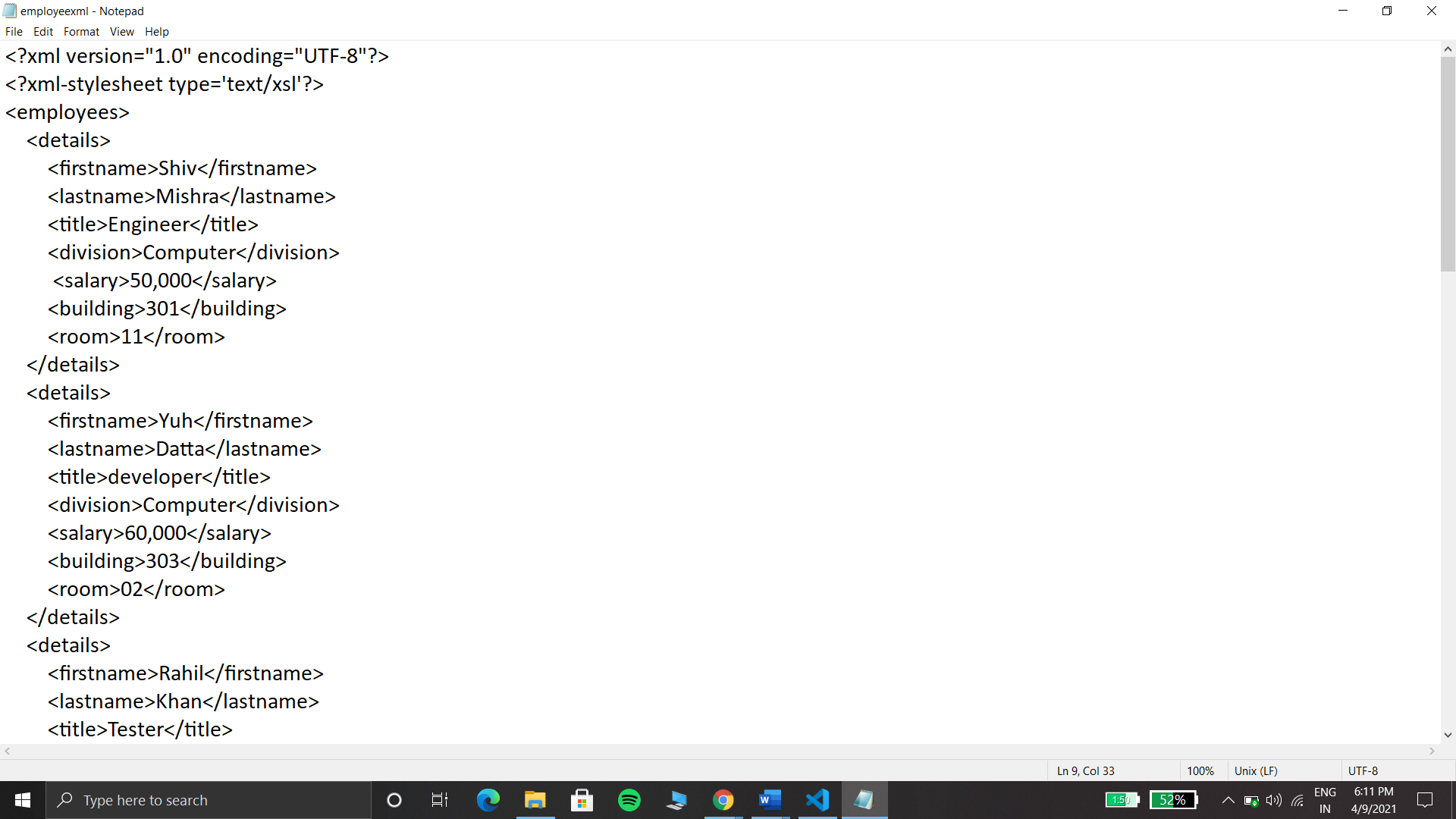
*Figure 12: Residents of a country json data*



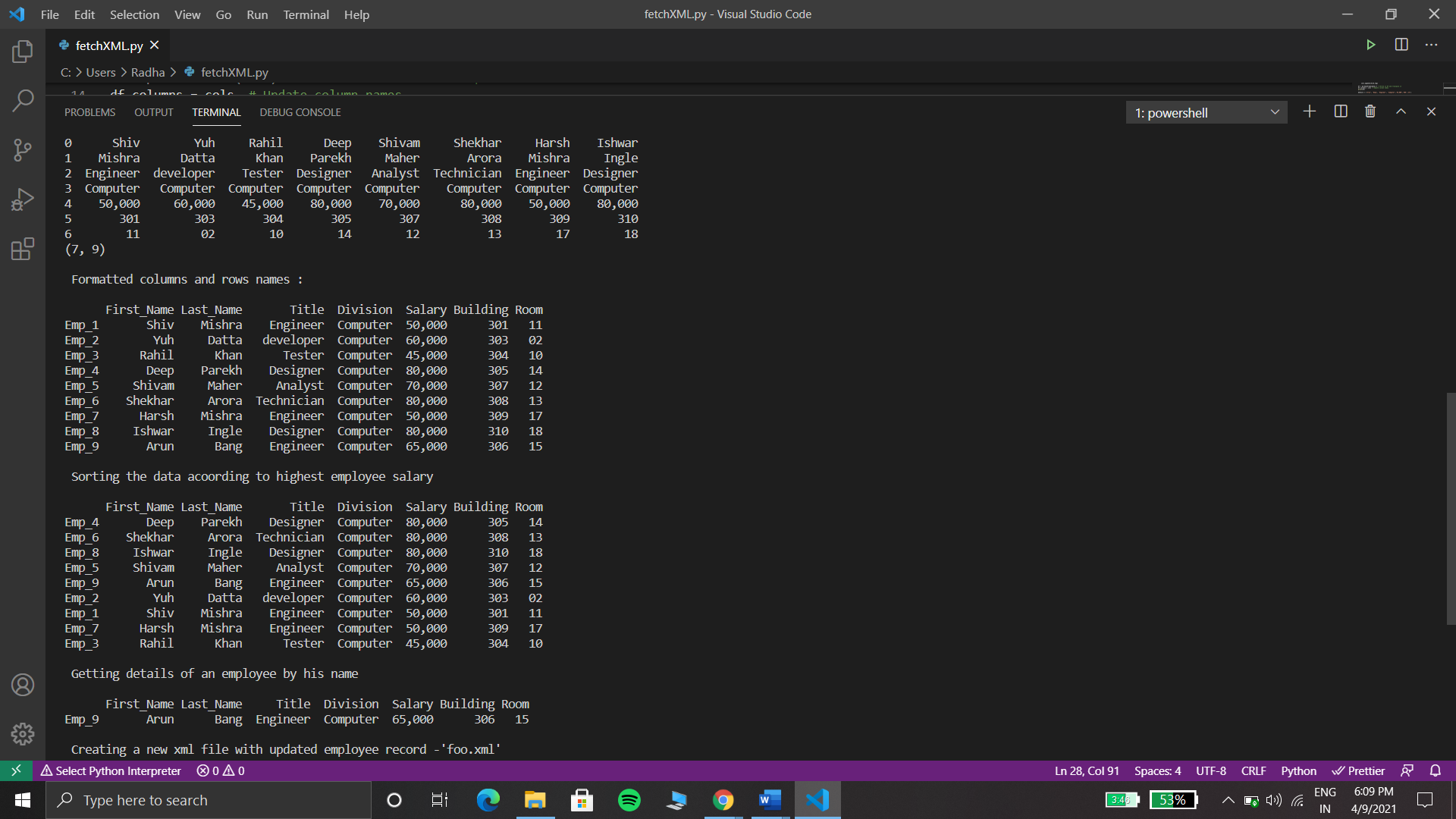
*Figure 13: Representation of json data in tabular form*

## Fetching data from xml file and displaying it using pandas:

* Pandas provide an effective solution for reading and traversing this large dataset files (xml/json/csv).
* Given the structure of XML files, we can represent them as tree, and this approach is used by the xml. etree. ElementTree python module.
* The parsing of the xml files starts at the root of the tree. Iterating to each node means one can get each element and all its sub-elements and can build the dataframe.
* In the below example of employee dataset, xml file is parsed, new employee is added, employees are sorted according to their salaries. For a proper tabular display of the data, the columns are renamed. A new xml file is created using the pandas’ function.

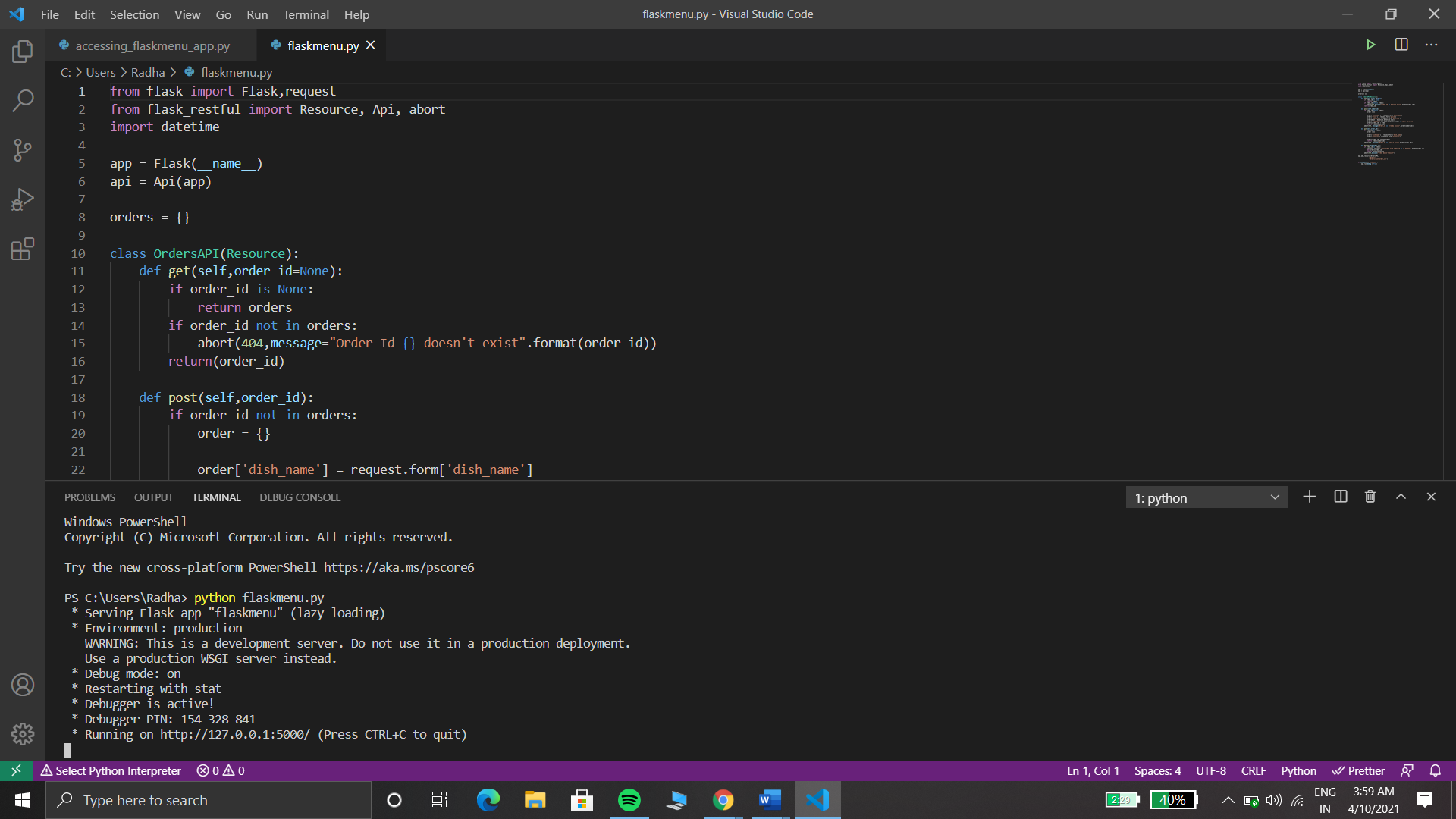


*Figure 14: Employee data xml file*

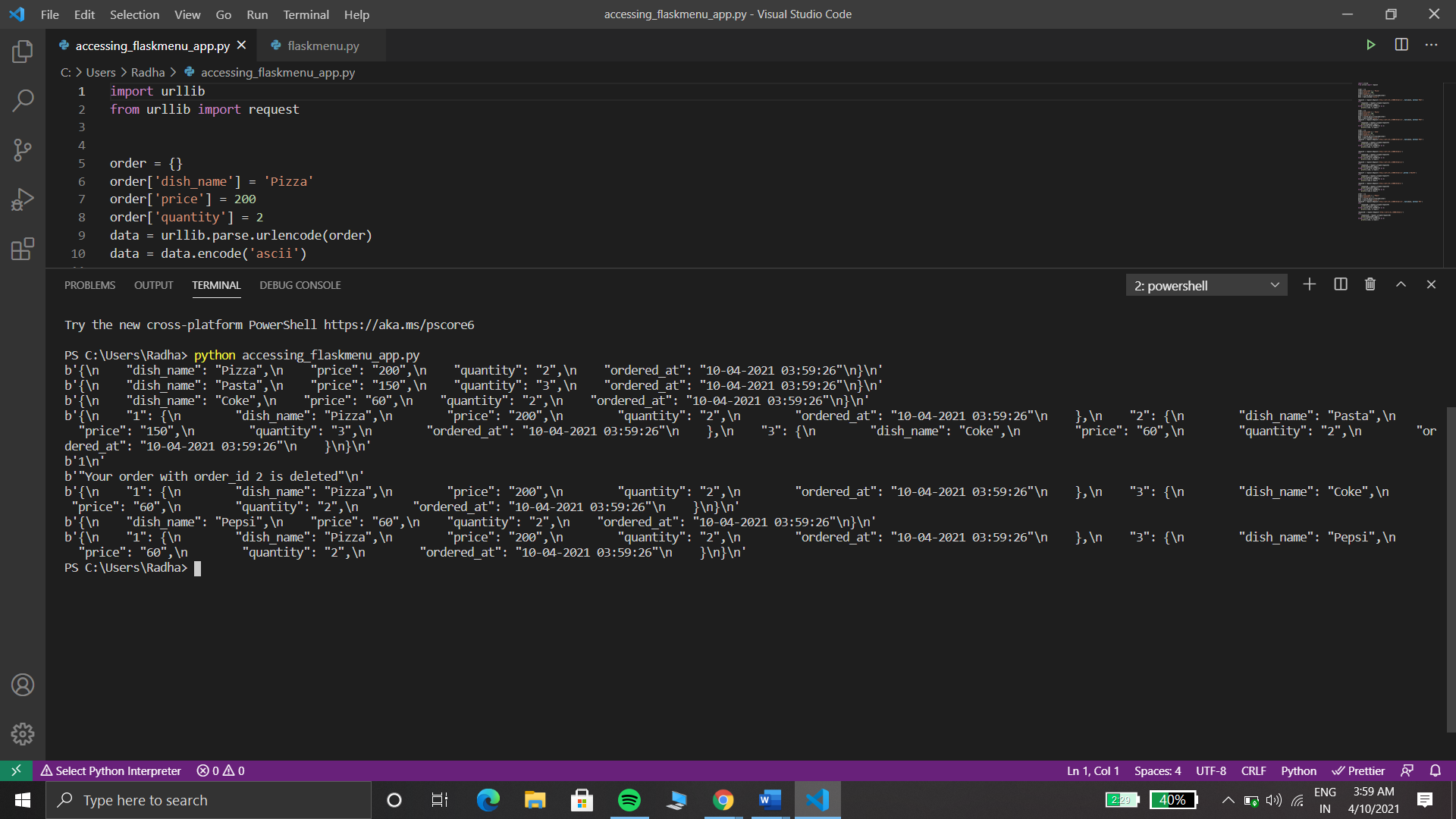
*Figure 15: Tabular representation of xml data.*

## Flask API for Post and Get Requests:

* Flask extensions help in building RESTful services with Flask. The clients of the web services will need to add, remove, modify and post tasks. These tasks need to be stored as a tasks list in the memory structure.
* To implement the REST APIs post,get, put and delete methods, I have created a food ordering flask service.
* The order of the user is available by the use of HTTP GET() method. The get method displays the attributes like what is ordered, the price of the dish ordered and the time at which the order is created.
* With the help of HTTP POST() method, a new order can be placed.
* The DELETE() methods removes the desired order.
* The PUT() method modifies the order.
* The request to carry out this methods are all associated with the URl. The flask get,put,delete and post functions are all defined in the class.
* Below is the snippet of server running and the response send by the server.



*Figure 16: Flask server*



*Figure* : *Response sent by the server.*

The link to the GitHub repository of the assignment –

[**https://github.com/Radha34/Web-Development-Assignment-**](https://github.com/Radha34/Web-Development-Assignment-)

# Learnings

## JavaScript-

* Being through with the html and CSS is not enough if you want to create some amazing website. Javascript and jQuery are the ones that add life to your site.
* JavaScript functions and object performs the major work. A JavaScript function is a block of code designed to perform a particular task. It is executed when "something" invokes it (calls it).
* JavaScript objects are containers for named values, called properties and methods.
* The localStorage is a property that allows JavaScript sites and apps to save key/value pairs in a web browser with no expiration date.
* To use localStorage in your web applications, there are five methods to choose from:

1. setItem (): Add key and value to localStorage.
2. getItem (): This is how you get items from localStorage.
3. removeItem (): Remove an item by key from localStorage.
4. clear (): Clear all localStorage.
5. key (): Passed a number to retrieve the key of a localStorage.

## Pandas-

* Pandas is a popular Python data analysis tool.
* It is used to analyze and manipulate data.
* It is used to perform tasks like data visualization, manipulation, data fill, statistical analysis, loading and saving of data.
* It provides easy to use and highly efficient data structures, Series and DataFrame.
* These data structures deal with numeric or labeled data, stored in the form of tables.
* A DataFrame is 2-D shaped and contains data of different parameters, captured from multiple observations.
* Each observation is represented by a single row, and each parameter by a single column.

A dataframe is created by loading the datasets from the csv, xml, json files or SQL databases. The type of a Pandas dataframe is either lists or dictionary or a list of dictionaries.

Here’s an example of creating a dataframe-

import pandas as pd

# list of strings

lst = ['DBMS', 'OS', 'DS', 'DAA', 'SS', 'CP', 'TOC']

# Calling DataFrame constructor on list

df = pd. DataFrame(lst)

print(df)

The output will be:

|  |  |
| --- | --- |
|  | 0 |
| 0 | DBMS |
| 1 | OS |
| 2 | DS |
| 3 | DAA |
| 4 | SS |
| 5 | CP |
| 6 | TOC |

The dataframes are used to manipulate the data. Once you updated/created the data, you can export them to csv, xml or json files.

## Flask-

* Flask is a micro web framework written in python. It does not require any tool or library.
* It has no database abstraction layer, form validation, or any other components where third party libraries provide common functions.
* It is used for web applications, API designing and working with third party APIs.
* One can easily extend flask to develop complex applications as well.

Here’s a simple helloworld flask app-

from flask import Flask

app = Flask(\_\_name\_\_)

@app. route ('/')

def hello\_world ():

return 'Hello, World!'

if \_\_name\_\_ == '\_\_main\_\_':

app. run ()

This will return a ‘Hello, World!’ to the server.