Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 3, June 2021: 2428-2438

Research Article

HTML Content and Cascading Tree Sheets: Overview of Improving Web Content Visualization

Apoorva Ganapathy[1], Siddhartha Vadlamudi[2], Alim Al Ayub Ahmed[3*], Md. Shakawat Hossain[4], Md. Aminul Islam[5]

ABSTRACT

The database system has contributed immensely to the present state of web pages and caching. Also, there are so many layers in broad-spectrum for catching numerous web content with culmination with extension .html, XML, .json, .txt, .jpg, .pdf, .png, .gif among others. Examples include web servers, content delivery networks, etc. But, these commonalities of presentational HTML cannot separate HTML content, which enables HTML documents harder to create, reuse, maintain, and tailor. Hence the objective of this study was aimed at cascading cache layer in a content management system using cascading tree sheets (CTS), an insubstantial language for unfolding the presentational aspects of page strategy that the commonalities of presentational HTML cannot characterize. To address the purpose of this study, we chose to address the implementation of cascading tree sheets and selection in the cascading cache layers in content management. The result showed the essential uncomplicatedness of cascading tree sheets by extracting its directions downcast to 2 tree processes that are related to the device and have helpful derivation preserving features.

Keywords: Cascading Cache Layer, CMS, Cascading Tree Sheets, HTML Content

INTRODUCTION

At the beginning of the 1960s, databanks structures have been a driving force to active websites, and caching and appearance have been the principal approaches to enhance the scalability, manageability, and performance of such network datasets. It differs from conventional datasets

[1]Senior Developer, Adobe Systems, San Jose, California, USA
[2]Software Engineer II, Xandr, AT&T Services Inc., New York, USA
[3]School of Accounting, Jiujiang University, 551 Qianjindonglu, Jiujiang, Jiangxi, China
[4]Department of Accounting & Information System, Jagannath University, Dhaka 1100, Bangladesh
[5]Department of Electrical & Electronic Engineering, East Delta University, Chattogram 4209, Bangladesh
*Corresponding Email: alim@jju.edu.cn

settings, the software elements of a web dataset, consisting of dataset, web, and application servers as well as other middleware which are mainly self-regulating from one another, despite working together as a complete system. Cascading cache layer brought to light the separation of the content

of an HTML document from the presentation. This evolvement enables easy maintenance of sites, share panache panes through pages, and modify pages to diverse locations (Ganapathy, 2019). Although an important portion of a contemporary web page construction is not supported by cascading cache layer, and are rather distinct by presentational HTML construction block and java scripts, interleaved with HTML content. HTML content is enveloped in layers of presentational HTML to offer support points for cascading cache layer's competencies. For instance, when considering the nodes, the mass of HTML, each allotment appearance fragment, that was required just to put content in a curved frame before the introduction of a distinct instruction for doing so.
This article presents cascading cache layer in a content management system using cascading tree sheets (CTS), an insubstantial language for unfolding the presentational aspects of page strategy that the commonalities of presentational HTML cannot characterize.

Problem Statement

There is lot of layers in broad-spectrum for catching numerous types of documents culmination with extension .html, XML, .json, .txt, .jpg, .pdf, .png, .gif among others. Examples include web servers, content delivery networks, etc. However, these commonalities of presentational HTML cannot separate HTML content, which enables HTML documents harder to create, reuse, maintain, and tailor. To overcome these aforementioned issues, cascading cache layer is introduced as a concept that not only caches files but then again allots distinctive keys to the files by adding a file in corresponding to the cached file stating down the direction of reliance with other files loaded before this file, so that during re-caching or cache invalidation, the sequence is followed so that the cache gets deleted according to the references (Ganapathy et al., 2021). This would circumvent a lot of needless cache invalidations and anchors cache re-use in turn accumulating the percentage of cache success so that the performance of the page remains highly improved.
Cascading cache layer will make authors capture presentational system just like cascading style sheet which makes them compress presentational flair. Cascading cache layer through cascading tree sheet pickers and syntax to insert presentational HTML against HTML content. Some parts of cascading tree sheet designed page that mesmerizes a beginner author can be inserted easily by such authors, networking to the tree sheet and fixing suitable categories on their HTML components. The cascading cache layer is of great benefit to beginners because it enables them to get the sheet deprived of requiring to comprehend in what manner it is executed. Also, as a result, the inserted HTML comprises Javascript that performs on the content, cascading tree sheet also offers a generic, simple manner for beginners to covering their HTML content in potent Javascript presentations quick to respond design or D3 visualizations deprived of sight on its line of Javascript. Cascading cache layer is more refined and its aids a website builder completely screen on HTML manuscript into its content as well as presentational splits. This enables content tranquil to control, spick, and span by presentational HTML, which can be recycled transversely pages and tailor-made to diverse settings. With this capacity to compress, and allot, the presentational system supports an amusing biota of content recycle for all layers of the website expansion biota (Ahmed et al., 2013). Beginners can interpret an HTML board with a cascading style sheet category that converts it into a Google Map. Technical know-how scripters can take the lead of theme and proper compression deprived of having to apply a content management system. For instance, an expert could alter or modify the design of every single publication on her website by modifying one line of HTML to relate to another tree sheet rather than strenuously changing the HTML for every publication. Also, power operators can offer web, simply by using the correct cascading style sheet

categories. Cascading cache layer other benefits which will explain in the residue of the article (Pavlo et al., 2009). These include-

Isolation of content, and presentation: Cascading cache layer finishes the isolation of presentation from content that cascading style sheet initiated, ensuring web presentation and content stress-free and easier to maintain, create, reuse or recycle, and tailor. For instance, a scripter may design a specific blob of presentation HTML immediately and also ensure the application of it on numerous pages by petitioning the suitable cascading tree sheet.

Mockup-determined scheme: Cascading cache layer empowers the web designers to define themselves in basic HTML instances instead of template languages. "A designer can even compose a tree sheet representing on a third-party web page, utilizing it as \template" despite its composer did not blueprint for this.

Javascript is short of Javascript: If a mockup consists of Javascript, it will be appealed robotically when contented is enfolded. This stretches beginner users the capability to appeal to amusing Javascript effects and picturing on their content deprived of ever sighted, much less script, on its line of Javascript.

Template and Data Scratching: Capsizing its recognizable performance, cascading cache layer directs can be utilized to isolate both the content and template from a webpage.

LITERATURE REVIEW

Research to increase the status of web contenting scripting commonly drops into 1 out of 4 classes; framework-driven, language-driven, higher-order interfaces, and intelligence creation. Several language-driven methods try to find to validate strategy requires into dominant dictionaries for recycle. An example is HTML5, which are language developments with execution to back them up over the former version. Amin & Vadlamudi (2021) buttressed Micro formats and numerous semantic web presents like FOAF (Brickley and Miller, 2010), and respectable relations (Azad et al., 2021) embed extra specific vocabularies surrounded by HTML. Other categories like HTML Microdata (Hickson and Hyatt, 2011), and (RDFa (Adida et al., 2008) try to find to offer manners to drive in extensible lexes in the interior of HTML. Instead a novel approach for expression offering a modest manner to link previous expressions prepared from ISON and HTML structure embedded methods that tend to handle issues related to the shared programming pattern of the web. A template that sails (Tatsubori and Suzumura, 2009) spontaneously pressed template procedures into the client-side. Syn tool (Benson et al., 2010a) reiterated a modest pattern for spontaneously synchronizing and continuing rational data on the client-side. Other contexts in the rear, given that programmed server-side tenacity of alterations that arise in the client-side (Cannon and Wohlstadter, 2010). Intellect production methods emerge or flood the web presently, helping the operator to better comprehend and act on it. Kits like WEB Crystal (Chang and Myer, 2012) and Fire crystal (Oney and Meyers, 2009) support the writer comprehend why a piece of a web page seems or acts as it does so that they can repurpose it.

Lastly, higher-order manipulation kits support the operator do work short of sliding into HTML. WYSIWYG or text editors are canonical instances, on the other hand, a more recent study has centered on retargeting, which is a general practice that up until lately was only done manually. Copy Styler (Doewes et al., 2021) interactively supports operators retarget total pages with a crossing point that places them side-by-side. Bricolage (Kumar et al., 2011) utilizes machine learning to do page scale retargeting spontaneously. However, cascading cache layer is believed to be beneficial another half to these methods. Cascading cache layer clarifications possibly will serve as an influential semantic indicator for other kits to integrate. Contexts might make use of

them for automated tenacity and editing proper. Intellectual production and retargeting kits might use them to offer UI signals or as future feedback. Bricolage might use cascading cache layer to explain the retargeting which was constructed so that they can be recycled in the upcoming years. Several authors want to derive benefits from abundant widget archives, on the other hand, the absence of Javascript know-how limits them from performing such tasks (Khan et al., 2021). Facts gathering from several chats with some bloggers shows that there is some scenario many writers utilize moderately heavy-weight Javascript backgrounds such as Exhibit (Huynh et al., 2007) just for the relatively minuscule property of categories HTML boards. Huynh and co-authors stated selecting Exhibit as a result of providing them a means to achieve this by a mere editing HTML, without requiring or making use of Javascript (Benson et al., 2010b). The D3 picturing archive is an alternative example (Bostock et al., 2011). D3 offers the skill of creating amazing web picturing and the task website has a studio of samples for users to copy, edit and use on their own. The task is difficult due to the requirement of skill or appropriate understanding of Javascript, JSON, SVG, and D3's distinct programming pattern. It is time-consuming even with an experienced Javascript programmer.

Cascading cache layer offers a common method that enables plain HTML a professional bond into an opulent widget entreaty, ensuring its accessibility to beginners. Operators are only required to connect to a tree sheet at that juncture spot up their HTML with the appropriate category names. The cascading cache layer thereafter covers this HTML with a bootstrapped for the doodad execution. The HTML a beginner makes available in this scenario is not just content to show, but also for variables to pass to the doodad like pin scenes for data or map for the bar chart.

STREAMLINING HTML IN CONTENT MANAGEMENT SYSTEM

Conventional web pages are packed with an HTML framework whose only goal is to make available a system that cascading style sheet utilizes to produce the page project. Immediately this project is completed, it vestiges moderately motionless. But the programmer can continue to update and modify the HTML content of the web page either by adding news articles, etc. as well as paddle through HTML design to complete this assignment.

In a scenario of buying a website template, which is a huge business on the net, the template will come as HTML files. At this juncture, the responsibility of the buyer to brand to his/her taste. But with cascading cache layer, the web template will arrive in 3 folds namely, a content document, a mockup, and a tree sheet that connects each of them. Any content management will be carried out within the confine of the content document, and any bug fixes or style updates will be handle by the mockup (Ahmed & Ganapathy, 2021). The intricacy storing to the operator relies on the framework to the content proportion of the document. Hence, for a widget that consists n content field, in such case cascading cache layer needs an average of n+1 DOM nodes to ensure the mapping for each field and a container to hold them when likened to a more subjective design-oriented number in a conventional HTML. For subsequent updates by a writer who chooses to shift to a diverse template require not to temper on their content document rather they will just use a new tree to plot their content into a new mockup.

Conventional Web Caching

Li et al. (2003) recommended adaptively choose pages to cache, scaling the reply duration, and the dissolution occurrence of the cached web pages. Owing to structure precise novelty threshold, Li and co-authors suggest a novel driven adaptive vibrant content caching system to guarantee that the delivered content is either new or not older than the threshold. The algorithm sustains

supervising the reply period and the length of the dissolution sequence that is the duration taken to verify the authenticity of all pages in the cache (Vadlamudi, 2021b). If the reaction period is longer than the length of the dissolution cycle, the number of each cached query class is upsurges to reduce the length of the reaction period. If the dissolution cycle is more than the query reaction duration, the number of cached query classes is reduced to cut down the dissolution cycle. Ensuring the response period nearby to the dissolution cycle in a balance point, new data is curtained.
Bhide et al. (2004) suggest that the proxy work out a moment to restore feature with each cached data piece for the pull method and registers template coherency requirement with each needs former history on relevant data, which are not the server projection method. Bhide and co-authors buttressed that the server estimation method needs former antiquity on pertinent data, which are not appropriate for web data that is extremely vibrant and intrinsically erratic. They suggest 2 pieces of machinery to combine drive-oriented client surveillance, Push or pull, or push and pull. They validate that both of them happen to be the different time-based coherency necessities, and are strong to botches, well-organized and walkable as well.

Cache Standby

Cao and Irani (1997) display a cost-oriented document standby system for proxy caching. The primary objective of their study was to consider both the document size and the cost and there include network deferment of recovering the document once more if it were not cached. Despite this study was concentrating on static content, but the idea can be related to caching in a typical sense such as caching vibrant data. Wolman (2002) emphasizes cache standby algorithm as a moderate unresponsive to web caching as a result of two hundred and twenty-six (226) practicing interrelated work.

Supportive Caching

Ramaswamy and Li (2004) research on supportive caching of web content according to the cessation period of each cache in the group. The cessation period of cached data shows the access argument in cache; Ramaswamy and co-author suggest a content placement system centered on this concept. The system offers a global organization of the whole disk size of all caches and successfully decreases copies cached devoid of degeneration activities (Ahmed, 2020). They add to their study the architectural design of a cache class for supportive web caching (Ramaswamy et al., 2005). Dynamic and hash-oriented procedures were developed in this investigation for content lookup and update within the class and new utility-oriented procedures for content assignment (Vadlamudi, 2020b). Ramaswamy and co-authors investigation influence automatic page fragmentation on web caching (Ramaswamy et al., 2005), and suggest 2 well-organized systems to segment a set of caches in many supportive classes to enhance the caching activities (Ramaswamy et al., 2006). Tang et al. (2004) express device assignment for many websites over a cache category as an enhancement tricky. Tang and co-authors suggest a vibrant programming solution to solve the enhancement issue, supposing the access rate of every device in individual cache is called precedence. Subsequently, all web devices to be cached require to be called before the caching resolutions are made, this method needs a robust server collaboration ability for the cache class (Ahmed et al., 2021).

Updates of Web Cache

Banga et al. (1997) suggest positive channels as incremental apprises for dormancy lessening over a dull network. Banga and co-author place a layer of proxies on both ends of a dull link. The server-

side proxy positively directs data (which are probably stale) to the client-side proxies throughout the futile time. Devising transported all the data just once, in the enduring communication, only a validation that the data are modified or a channel, which is the alteration among the outdated version and the present one, will be conveyed (Donepudi et al., 2020). Mogul and co-authors further measure the advantages of channel encoding utilizing real smidgens. Their results exhibit incremental updates offering significant growths in the response size and delay. Also, make available data firmness benefits, and synchronization of channel encoding and data firmness produces the best outcomes (Mogul et al., 1997).

METHODS

In an attempt to discourse the purpose of this study, which is cascading cache layer in content management system using cascading tree sheets (CTS), a trivial language for unfolding the presentational aspects of page strategy that the commonalities of presentational HTML cannot characterize. We decided to discuss cascading cache layer through cascading tree sheets implementation and the selections under this section.

CASCADING TREE SHEET IMPLEMENTATION

All cascading tree sheet commands can be conveyed in 2 fundamental terms, namely; map and graft. A graft connection denotes a correspondence among 2 components whereas a map connection denotes a correspondence amongst 2 combinations of components (see Figure 1). These associations may be assumed to be directionless edges that link nodes on 2 fixed trees. Map and graft could be examined in both directions (Ganapathy, 2019). In term of mockup compelled expansion, a direction-map direct the content to mockup, as the second channel overwrites the content with mockup quality. 'In robust templating, one channel combines JSON data into the template, while the other channel jams JSON data back out of a template'

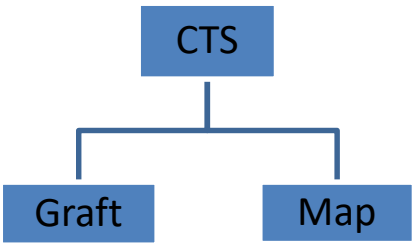

Figure 1: Fundamental term of cascading tree sheet

Selections

A selector is a tree probe that when examined in contrast to a tree, proceeds an assortment. The commonality caching selector can be used to generate an HTML allotment, and a key path selector can be utilized to generate a JSON assortment, for instance, conventionally, jquery – we assume an allotment as a group of nodes $S_{\text{commonality cache}} = \{n_1, n_2,..n_n\}$. A cascading tree sheet allotment is rather a set of classes, where the individual group is a set of genealogical nodes of the same source. To transform a commonality caching into cascading cache layer selection, the individual node allotted is merely converted into a class of size 1.

$S_{cts} = \{g_1, g_2....g_n\}$; $g_i = (n_i^1, n_i^2....\}$

This long-drawn-out concept of assortment is imperative for the reason that several real-world HTML assemblies do not enfold individual semantic units in a discrete DOM node. In a listing of files on a commercial website, for instance, each customary of 3 table rows may signify 1 volume. The HTML5 Microdata requirement looks to disregard this demonstrating tricky when discoursing its item range user (Vadlamudi, 2021a). For the reason that tr must be the teen of table or tbody, for instance, it is incredible to definite such a multi-node entry possibility with Microdata (Ganapathy et al., 2020). The cascading tree sheet assortment ideal consents us to switch these circumstances, on the condition that the selector syntax offers a method to define it. Table 1 summarizes graft and map relation asserting 2 equivalent classes. Thus, one might substitute the other for the other.

TABLE 1
Summary of two fundamental items in cascading tree sheet

Graft and Is	A graft connection denotes a correspondence among 2 components (see Figure 2).
	This is regulation carries out a graft on a transformed assortment in which an individual node in the original assortment has been replaced by a class involving its clusters. Is connection may hence be assumed as graft with compulsory vessel contents semantics that makes it possible for the node to take part in the cascading tree sheet relation without alteration when it is operated. This signifies that armed with only the result of a cascading tree sheet process can be recognized, and their inner HTML characterizes the connected tree portion from the other tree.
MAP and ARE	MAP and ARE go after the same model in that an ARE is a MAP with vessel semantics fastened on. They either stimulate the iterative loop seen in programming languages. When carried out directionally as a process, MAP acts as a Functor that functions on either connected trees or the cascading tree sheet relatives express between them.

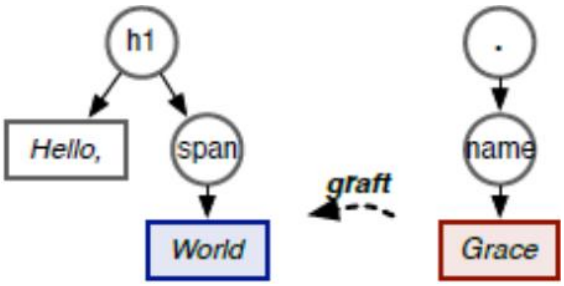

Figure 2: Graft Operation template

RESULTS AND DISCUSSION

This section discusses the result from the question that we considered to assist us in addressing the aim of this study.
Can doodads introduce other doodads?

Yes, a doodad can introduce another doodad if the language such as cascading tree sheet which is a type of cascading cache layer is adopted, it is expected as this will be a very powerful property to arrange and reuse web content.

How about the main content?

From the literature review and other related work, cascading caching layer (cascading tree sheet) does not support any modification or editing of the main content that is heading in a web page (Paruchuri, 2019). The main is not in support of any suppression, which is the IS and ARE operators depend on. This implies a tree sheet only is not enough to lend the content, you have to relate it to the remote cascading and Javascript. Meanwhile, a Javascript-laden cascading tree sheet doodad could always add these commonality caching and javascript connects for you.

Is there any similarity between cascading tree sheet and XSLT?

Cascading tree sheet differs from XSLT on numerous faces. Cascading tree sheet explains directionless relation among 2 trees instead of modification from one tree to alternative, a tree sheet can be run in both ways. Also, XSLT demands stating the resulting web page in terms of an XML software design language. Cascading tree sheet rather utilizes a mockup compelled expansion method, which outcome in items that can be more willingly be imitated and reused. Lastly, XSLT botch semantics is to smash on misshapen feedback, which is not genuine for the web. It booms on the sympathetic of unpremeditated, copy-paste-brand software design that often consequences in slight mistakes (Paruchuri et al., 2021). Cascading tree sheet accepts Cascading style sheet letdown semantics and restricts and accounts letdowns in its place of hitting.

Is cascading tree sheet the same with HTML5 Web elements and Follower DOM specifications?

Cascading tree sheet is a general-purpose, outwardly quantified, tree relative language with a mere syntax (Vadlamudi, 2020a). This philological can be cast off to define templating, scraping, transcends, doodads, and mechanical compression. The Web Element and Follower DOM specifications are HTML-specification syntax delays to offer compression and reprocess HTML widgets. The Follower DOM specification makes a distinct class of node whose \tail" subtree is protected from the rest of the page in convinced methods, averting, for instance, cascading style sheet name galaxy crashes that can arise when doodads from diverse archives are co-mixed on a page.

Cascading tree sheet doodad operators would bennet importantly from this requirement being assumed for the reason that of these stouter compression assurances (Paruchuri, 2021). The Web Machineries specification makes available an HTML syntax for stating and recycling web doodads (Vadlamudi et al., 2021). Like cascading tree sheet, it also offers a way to map content into new doodad cases, however, these mappings are restricted to HTML, while cascading tree sheet can graft with JSON (applied) or any tree-system data (hypothetical). Web Technologies also run down doodad descriptions and mapping directions with the HTML content. This averts mockup-driven and expostfacto doodad and subject formation.

CONCLUSION

This article has shown cascading cache layer in content management system through cascading tree sheets, a suggestion, and execution of a trivial language that offers the same type of compression for HTML smartness. We show usage situations with working executions showing how cascading tree sheets can increase a wide-ranging of writing responsibilities for novices, web designers, and archive writers. Lastly, we showed the essential uncomplicatedness of cascading tree sheets by extracting its directions downcast to 2 tree processes that are related to the device and have helpful derivation preserving features.

REFERENCES

1. Ahmed, A. A. A. (2020). Corporate attributes and disclosure of accounting information: Evidence from the big five banks of China. J Public Affairs. e2244. https://doi.org/10.1002/pa.2244
2. Ahmed, A. A. A., & Ganapathy, A. (2021). Creation of Automated Content with Embedded Artificial Intelligence: A Study on Learning Management System for Educational Entrepreneurship. Academy of Entrepreneurship Journal, 27(3), 1-10, https://doi.org/10.5281/zenodo.4973057
3. Ahmed, A. A. A., Siddique, M. N., & Masum, A. A. (2013). Online Library Adoption in Bangladesh: An Empirical Study. 2013 Fourth International Conference on e-Learning "Best Practices in Management, Design and Development of e-Courses: Standards of Excellence and Creativity", Manama, 216-219. https://doi.org/10.1109/ECONF.2013.30
4. Ahmed, A. A. A.; Paruchuri, H.; Vadlamudi, S.; & Ganapathy, A. (2021). Cryptography in Financial Markets: Potential Channels for Future Financial Stability. Academy of Accounting and Financial Studies Journal, 25(4), 1–9. https://doi.org/10.5281/zenodo.4774829
5. Amin, R., & Vadlamudi, S. (2021). Opportunities and Challenges of Data Migration in Cloud. *Engineering International*, *9*(1), 41-50. https://doi.org/10.18034/ei.v9i1.529
6. Azad, M. M., Ganapathy, A., Vadlamudi, S., Paruchuri, H. (2021). Medical Diagnosis using Deep Learning Techniques: A Research Survey. *Annals of the Romanian Society for Cell Biology*, *25*(6), 5591–5600. Retrieved from https://www.annalsofrscb.ro/index.php/journal/article/view/6577
7. Banga, G. Douglis, F. and Rabinovich, M. (1997). Optimistic deltas for WWW latency reduction, in USENIX Annual Technical Conference.
8. Benson, E. Marcus, A. Karger, D. and Madden, S. (2010a). Sync kit: a persistent client-side database caching toolkit for data intensive websites. In WWW.
9. Benson, E., Marcus, A. Howahl, F. and Karger. D. (2010b). Talking about data: sharing richly structured information through blogs and wikis. In ISWC.
10. Bostock, M. Ogievetsky, V. and Heer. J. (2011). D3: Data-Driven Documents. IEEE Transactions on Visualization and Computer Graphics.
11. Brickley D. and Miller. L. (2010). FOAF Vocabulary Specication 0.98. Namespace Document.
12. Cannon B. and Wohlstadter. E. (2010). Automated Object Persistence for JavaScript. In WWW.
13. Cao P. and Irani, S. (1997). Cost-aware WWW proxy caching algorithms, in USENIX Symposium on Internet Technologies and Systems.
14. Chang, K. S.-P. and Myers. B. A. (2012). WebCrystal: understanding and reusing examples in web authoring. In CHI.
15. Doewes, R. I.; Ahmed, A. A. A.; Bhagat, A.; Nair, R.; Donepudi, P. K.; Goon, S.; Jain, V.; Gupta, S.; Rathore, N. K.; Jain, N. K. (2021). A regression analysis based system for sentiment analysis and a method thereof. Australian Official Journal of Patents, 35(17), Patent number: 2021101792. https://lnkd.in/gwsbbXa
16. Donepudi, P. K., Banu, M. H., Khan, W., Neogy, T. K., Asadullah, ABM., & Ahmed, A. A. A. (2020). Artificial Intelligence and Machine Learning in Treasury Management: A Systematic Literature Review. International Journal of Management, 11(11), 13–22. https://doi.org/10.5281/zenodo.4247297

17. Ganapathy, A. (2019). Image Association to URLs across CMS Websites with Unique Watermark Signatures to Identify Who Owns the Camera. *American Journal of Trade and Policy*, *6*(3), 101-106. https://doi.org/10.18034/ajtp.v6i3.543
18. Ganapathy, A. (2019). Mobile Remote Content Feed Editing in Content Management System. *Engineering International*, *7*(2), 85-94. https://doi.org/10.18034/ei.v7i2.545
19. Ganapathy, A., Ahmed, A. A. A, Siddique, M. NEA., (2021). Easy URLs in the Content Management System with Crawlers for Added Security. Academy of Marketing Studies Journal, 25(4), 1-10. https://doi.org/10.5281/zenodo.5002945
20. Ganapathy, A., Redwanuzzaman, M., Rahaman, M. M., & Khan, W. (2020). Artificial Intelligence Driven Crypto Currencies. *Global Disclosure of Economics and Business*, *9*(2), 107-118. https://doi.org/10.18034/gdeb.v9i2.557
21. Hickson, I. and Hyatt, D. (2011). HTML5. W3C Working Draft.
22. Huynh, D. F., Karger, D. R. and Miller, R. C. (2007). Exhibit: lightweight structured data publishing. In WWW.
23. Khan, W., Ahmed, A. A. A., Vadlamudi, S., Paruchuri, H., Ganapathy, A. (2021). Machine Moderators in Content Management System Details: Essentials for IoT Entrepreneurs. Academy of Entrepreneurship Journa, 27(3), 1-11. https://doi.org/10.5281/zenodo.4972587
24. Kumar, R., Talton, J. O., Ahmad, S. and Klemmer, S. R. (2011). Bricolage: example-based retargeting for web design. In CHI.
25. Li, W.S., Po, O., Hsiung, W.P., Candan, K. S. and Agrawal, D. (2003). Freshnessdriven adaptive caching for dynamic content web sites. Data & Knowledge Engineering, vol. 47, no. 2, pp. 269–296.
26. Mogul, J. C., Douglis, F., Feldmann, A. and Krishnamurthy, B. (1997). Potential benefits of delta encoding and data compression for http, in SIGCOMM, pp. 181–194.
27. Oney, S. and Myers, B. (2009). Fire Crystal: Understanding interactive behaviors in dynamic web pages. In VLHCC.
28. Paruchuri, H. (2019). Market Segmentation, Targeting, and Positioning Using Machine Learning. *Asian Journal of Applied Science and Engineering*, *8*(1), 7-14. Retrieved from https://journals.abc.us.org/index.php/ajase/article/view/1193
29. Paruchuri, H. (2021). Conceptualization of Machine Learning in Economic Forecasting. *Asian Business Review*, *11*(1), 51-58. https://doi.org/10.18034/abr.v11i1.532
30. Paruchuri, H.; Vadlamudi, S.; Ahmed, A. A. A.; Eid, W.; Donepudi, P. K. (2021). Product Reviews Sentiment Analysis using Machine Learning: A Systematic Literature Review. *Turkish Journal of Physiotherapy and Rehabilitation*, *23*(2), 2362-2368, https://turkjphysiotherrehabil.org/pub/pdf/322/32-2-316.pdf
31. Pavlo, A. Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S. and Stonebraker, M. (2009). A comparison of approaches to large-scale data analysis, in SIGMOD'09: Proceedings of the 35th SIGMOD International Conference on Management of Data, pp. 165–178, New York, NY, USA: ACM.
32. Ramaswamy L. and Liu, L. (2004). An expiration age-based document placement scheme for cooperative web caching, IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 5, pp. 585–600.
33. Ramaswamy, L. Liu, L. and Zhang, J. (2006). Efficient formation of edge cache groups for dynamic content delivery, in ICDCS, p. 43.

34. Ramaswamy, L. Iyengar, A. Liu, L. and Douglis, F. (2005). Automatic fragment detection in dynamic web pages and its impact on caching. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 859–874.
35. Tang, X. Chanson, S. T. Chi, H. and Lin, C. (2004). Session-affinity aware request allocation for web clusters, in ICDCS, pp. 142–149.
36. Tatsubori, M. and Suzumura, T. (2009). HTML Templates that Fly: A Template Engine Approach to Automated Oloading from Server to Client. In WWW.
37. Vadlamudi, S. (2020a). Internet of Things (IoT) in Agriculture: The Idea of Making the Fields Talk. *Engineering International*, *8*(2), 87-100. https://doi.org/10.18034/ei.v8i2.522
38. Vadlamudi, S. (2020b). The Impacts of Machine Learning in Financial Crisis Prediction. *Asian Business Review*, *10*(3), 171-176. https://doi.org/10.18034/abr.v10i3.528
39. Vadlamudi, S. (2021a). The Economics of Internet of Things: An Information Market System. *Asian Business Review*, *11*(1), 35-40. https://doi.org/10.18034/abr.v11i1.523
40. Vadlamudi, S. (2021b). The Internet of Things (IoT) and Social Interaction: Influence of Source Attribution and Human Specialization. *Engineering International*, *9*(1), 17-28. https://doi.org/10.18034/ei.v9i1.526
41. Vadlamudi, S., Islam, M. A., Hossain, M. S., Ahmed, A. A. A., Asadullah, ABM. (2021). Watermarking Techniques for Royalty Accounts in Content Management Websites for IoT Image Association. Academy of Marketing Studies Journal, 25(4), 1-9. https://doi.org/10.5281/zenodo.5002840
42. Wolman, A. (2002). Sharing and caching characteristics of internet content, Ph.D. thesis, University of Washington.