Training Seminar (MC-403)

DELHI TECHNOLOGICAL UNIVERSITY (DTU)

ANISH SACHDEVA

DTU/2K16/MC/13

Full Stack Software Developer at CERN, Geneva, Switzerland

Role: Full Stack Software Developer

Organization: CERN, Geneva, Switzerland

Department: FAP-AIS-GI

Supervisor: Mr. Jan Janke

Duration: 1st September 2018 – 31st October 2019 (14 Months)

[Attestation Letter Link]()

Summer Research Fellow @ University of Auckland, New Zealand

Role: Research Fellow in the Field of Geometry of Mobius Transformations

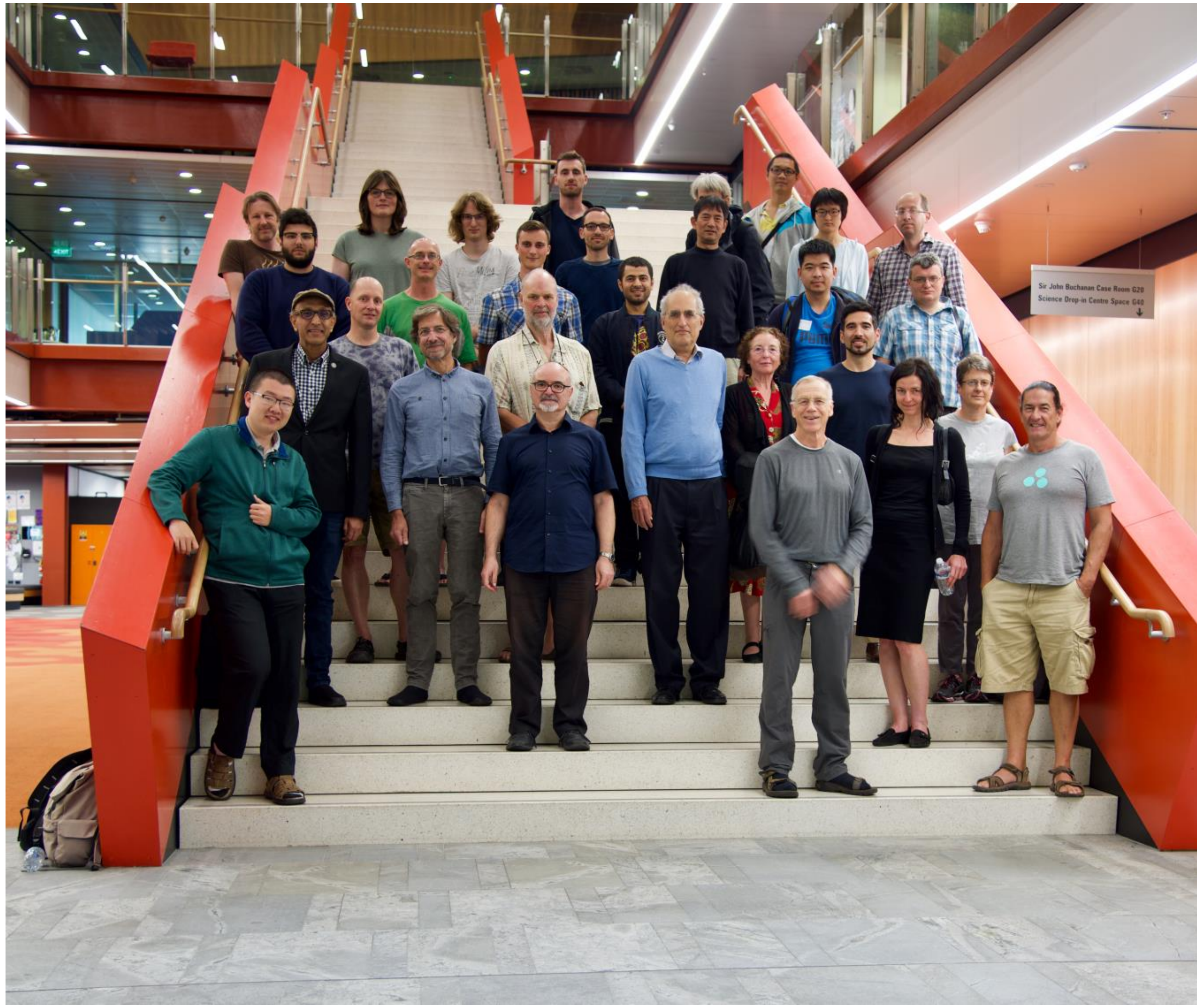
Organization: University of Auckland, New Zealand

Department: Dpt. Of Pure Mathematics

Supervisor: Dr. Pedram Hekmati

Duration: Jan – Feb 2020

[Invitation Letter Link]


Sir John Buchanan Case Room G20
Science Drop-in Centre Space G40

MITACS Globalink Research Scholarship

Role: Researcher in Computer Vision

Organization: University of Athabasca, Edmonton, Alberta, Canada

Sponsor: MITACS, Canada

Department: Department of Computer Science

Supervisor: Dr. Maiga Chang

Duration: June - August 2020

Project: Subtext: Creating a Context Based Real Time Communication Application

Project Link   Research Fellowship Award Letter

Subtext: A Context Based Real Time Communication Application

Training Seminar (MC-403)

ANISH SACHDEVA

DELHI TECHNOLOGICAL UNIVERSITY (DTU)

DTU/2K16/MC/13

Details

- Project Title: Subtext - Creating a Context Based Communication Application
- Subject: Training Seminar (MC-403)
- Research Institute: Athabasca University, Edmonton, Alberta, Canada
- Scholarship Award: MITACS Globalink Research Internship Scholarship
- Host Professor: Dr. Maiga Chang
- Submitted To: Dr. L. N. Das
- Candidate Name: Anish Sachdeva
- Candidate Roll No: DTU/2K16/MC/013
- Project Link: https://github.com/anishLearnsToCode/subtext

Frameworks and Stacks Learnt and Introduced To In This Project

- Angular 2+
- Node.js
- npm
- Git and GitHub + Version Control
- Firebase RealTime Database
- Firebase Console
- Markdown
- LaTeX
- Parsing and Regex

Aim: Things That I wanted to support

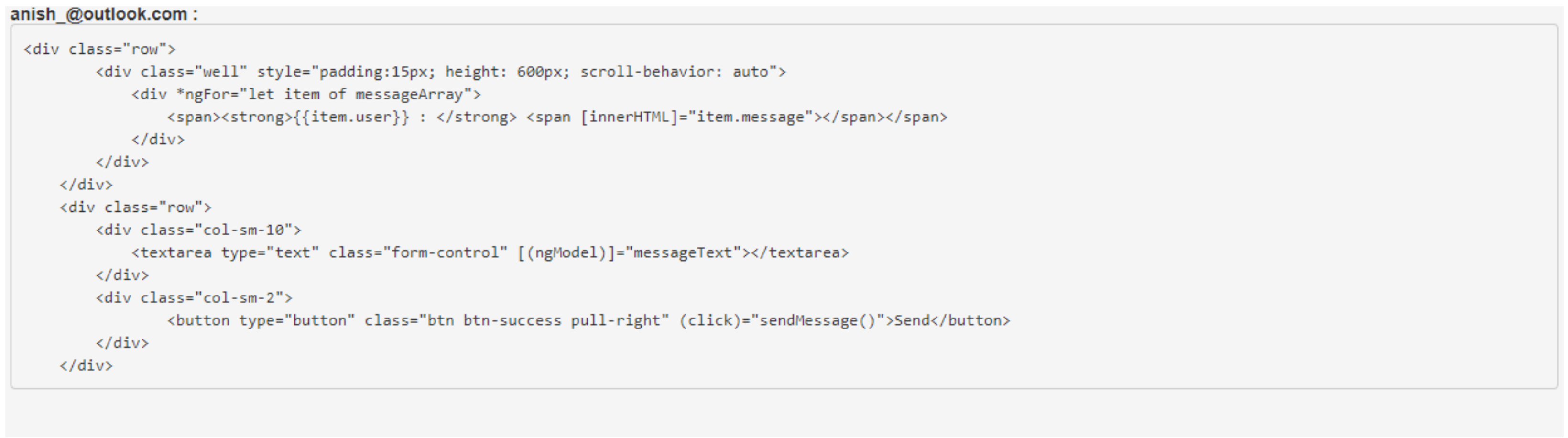
- Text
- Code
- Markdown
- LaTeX
- Emojis

Text

- Messages like this
- And this
- Also numbers 123456
- and small case + CAPITAL
- Plus special characters /*,.<>;'":}{][!@#$%^&*()_+=-

Java
C/C++
C#
JavaScript
Swift
Python
….
….
And everything in between

```java
class BinarySearchExample{
 public static void binarySearch(int arr[], int first, int last, int key){
   int mid = (first + last)/2;
   while( first <= last ){
      if ( arr[mid] < key ){
        first = mid + 1;
      }else if ( arr[mid] == key ){
        System.out.println("Element is found at index: " + mid);
        break;
      }else{
         last = mid - 1;
      }
      mid = (first + last)/2;
   }
   if ( first > last ){
      System.out.println("Element is not found!");
   }
 }
 public static void main(String args[]){
        int arr[] = {10,20,30,40,50};
        int key = 30;
        int last=arr.length-1;
        binarySearch(arr,0,last,key);
 }
}
```



anish_@outlook.com :

```html
<div class="row">
        <div class="well" style="padding:15px; height: 600px; scroll-behavior: auto">
            <div *ngFor="let item of messageArray">
                <span><strong>{{item.user}} : </strong> <span [innerHTML]="item.message"></span></span>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-sm-10">
            <textarea type="text" class="form-control" [(ngModel)]="messageText"></textarea>
        </div>
        <div class="col-sm-2">
                <button type="button" class="btn btn-success pull-right" (click)="sendMessage()">Send</button>
        </div>
    </div>
```

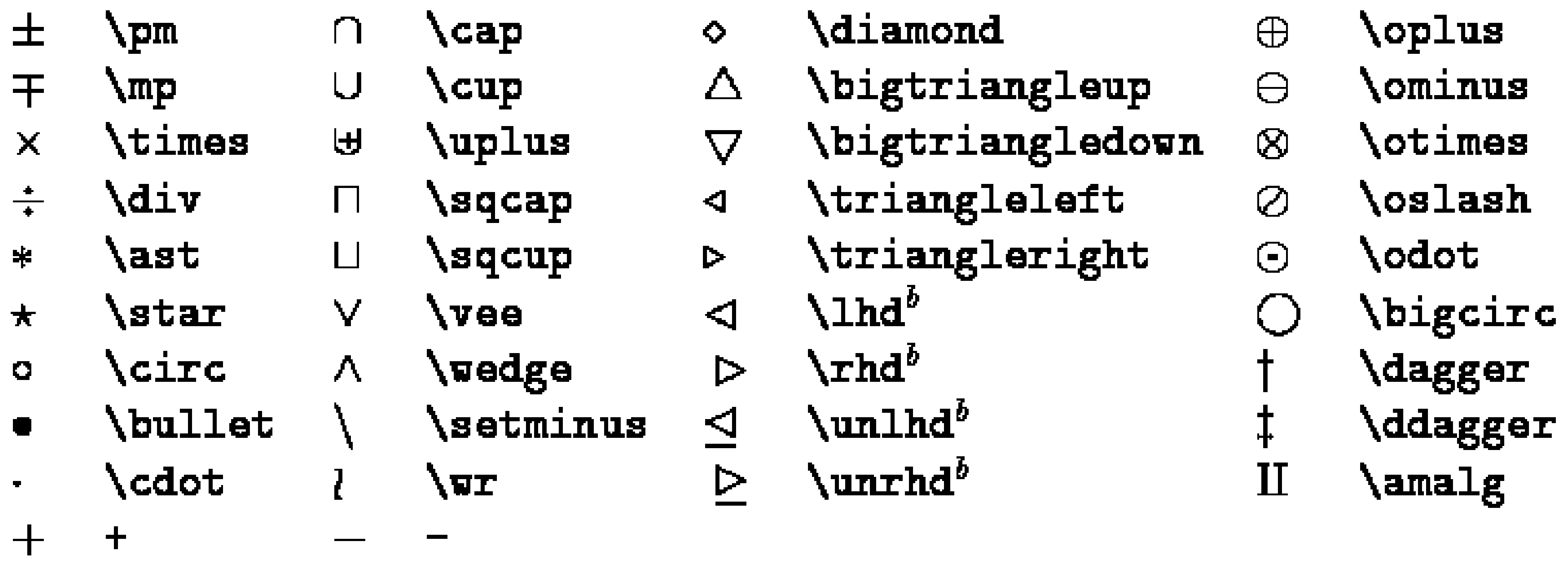
Markdown

- __bold__ --> **bold**
- _italic_ --> *italics*
- `code` -> code
- ```js``` --> perfomatted code
- # heading -> Main Heading etc.

Emojis

- 😀 😁 😂 🤣 😃 😄 😅 😆 😉 😊 😋 😎 😍 😘 ⍰ 😗 😙 😚 ☺ 🙂 🤗 ⍰ 🤔 ⍰ 😐 😑 😶 🙄 😏 😣 😥 😮 🤐 😯 😪 😫 😴 😌 😛 😜 😝 🤤 😒 😓 😔 😕 🙃 🤑 😲 ☹ 🙁 😖 😞 😟 😤 😢 😭 😦 😧 😨 😩 ⍰ 😬 😰 😱 ⍰ ⍰ 😳 ⍰ 😵 😡 😠 ⍰ 😷 🤒 🤕 🤢 ⍰ 🤧 😇 🤠 🤡 ⍰ ⍰ ⍰ 🤥 ⍰ ⍰ ⍰ 🤓 😈 👿 👹 👺 💀 👻 👽 🤖 💩 😺 😸 😹 😻 😼 😽 🙀 😿 😾

LaTeX



$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\bigtriangleup$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\bigtriangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$\ast$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\lhd$	\lhd^{b}	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\rhd$	\rhd^{b}	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\unlhd$	\unlhd^{b}	$\ddagger$	\ddagger
$\cdot$	\cdot	$\wr$	\wr	$\unrhd$	\unrhd^{b}	$\amalg$	\amalg
$+$	+	$-$	-				

Context based real time communication app that can understand information encoded with different text markup formats

$\downarrow$

SUBTEXT

Current Market and Competition

- Whatsapp
- Facebook Messenger
- Instagram
- Google Allo/Messenger
- Microsoft Office
- Skype
- Google Docs/Sheets/Slides

Text + Emojis

- Whatsapp
- Instagram
- Facebook Messenger
- Google
- Microsoft etc.

Text + Emojis + Markdown

- Slack
- Whatsapp (Italic and bold from Markdown)

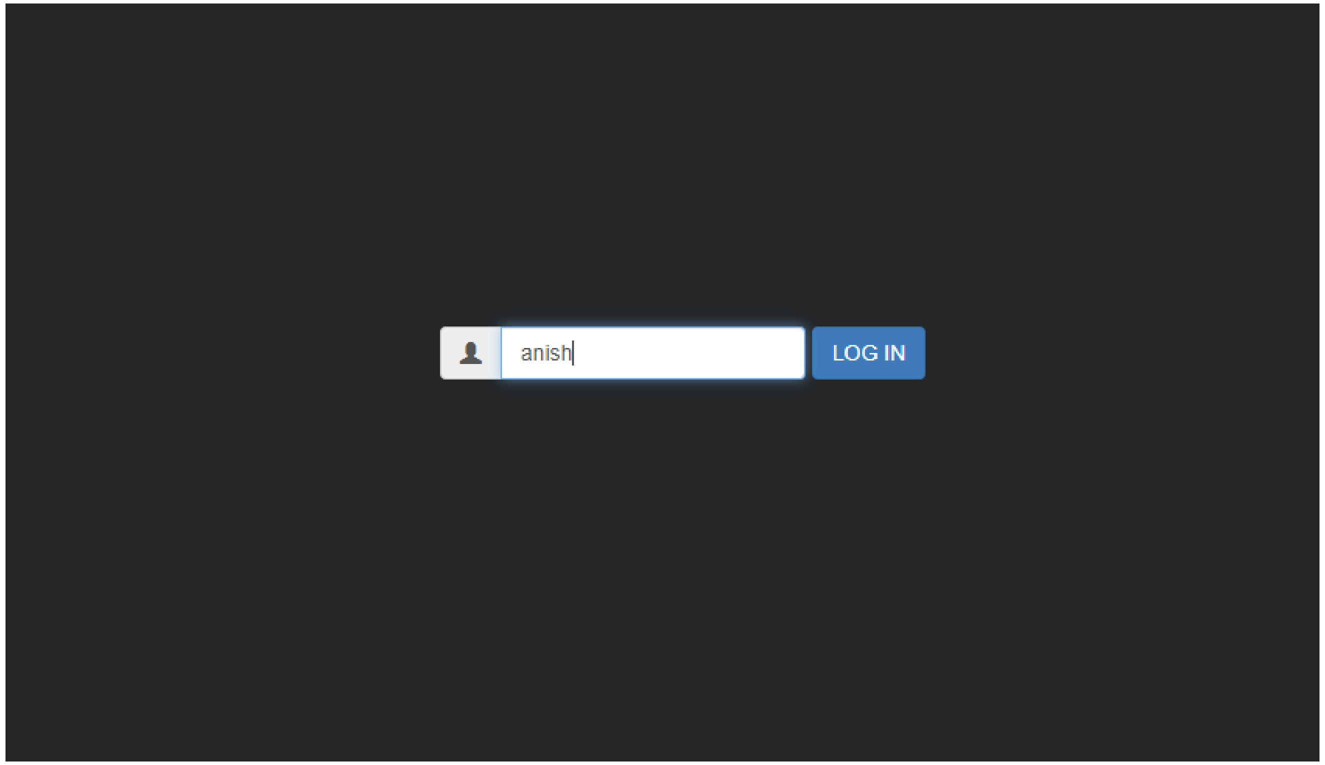
LaTeX + Others

None to my knowledge 🤷‍♀️

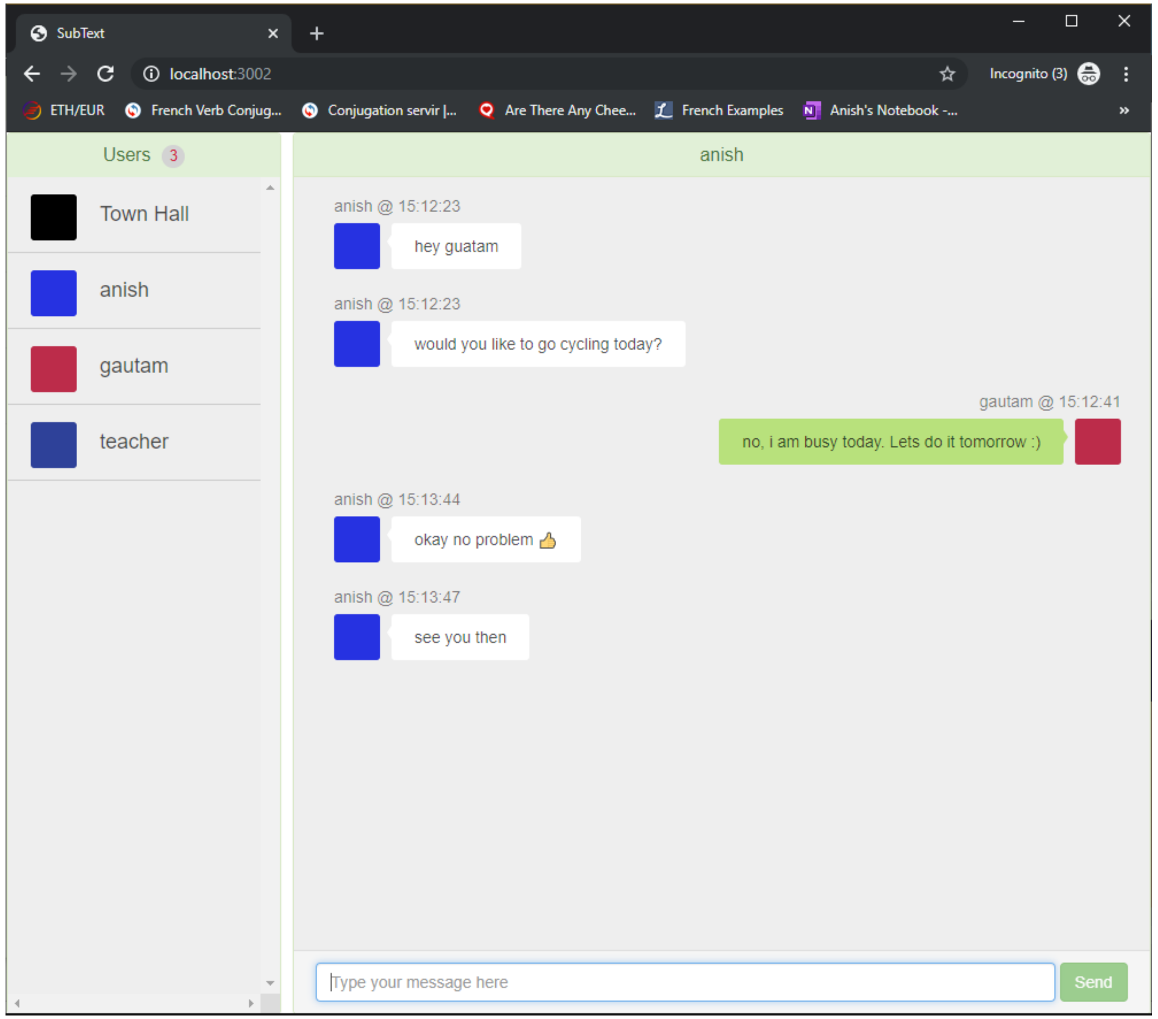
Initial Prototype

- Created using AngularJS, Express and MongoDB
- Has proper log in screen and text messaging facility people to people and also group chatting
- Also added chatbot that announces all active participants and when they leave the group etc.
- Disadvantages: does not support Markdown/LaTeX
- Another Disadvantage: was built on complete server side technology hence no client side rendering and computation available

Prototype I: Login page



Prototype I: Chatting Person/Person



Users 3

Town Hall

anish

gautam

teacher

anish

anish @ 15:12:23

hey guatam

anish @ 15:12:23

would you like to go cycling today?

gautam @ 15:12:41

no, i am busy today. Lets do it tomorrow :)

anish @ 15:13:44

okay no problem 👍

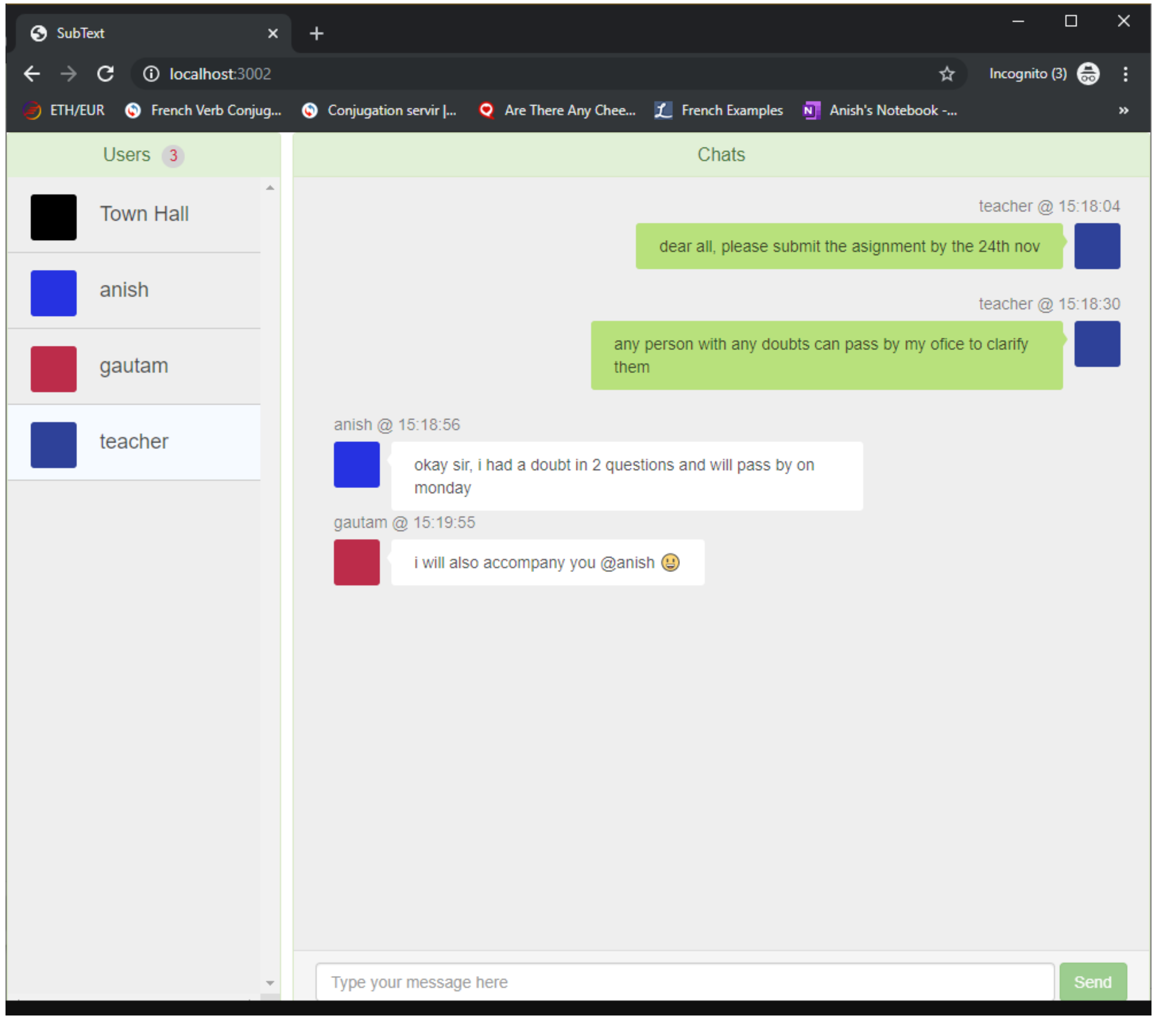
anish @ 15:13:47

see you then

Type your message here

Send

Prototype I: Person/Group



teacher @ 15:18:04
dear all, please submit the asignment by the 24th nov

teacher @ 15:18:30
any person with any doubts can pass by my ofice to clarify them

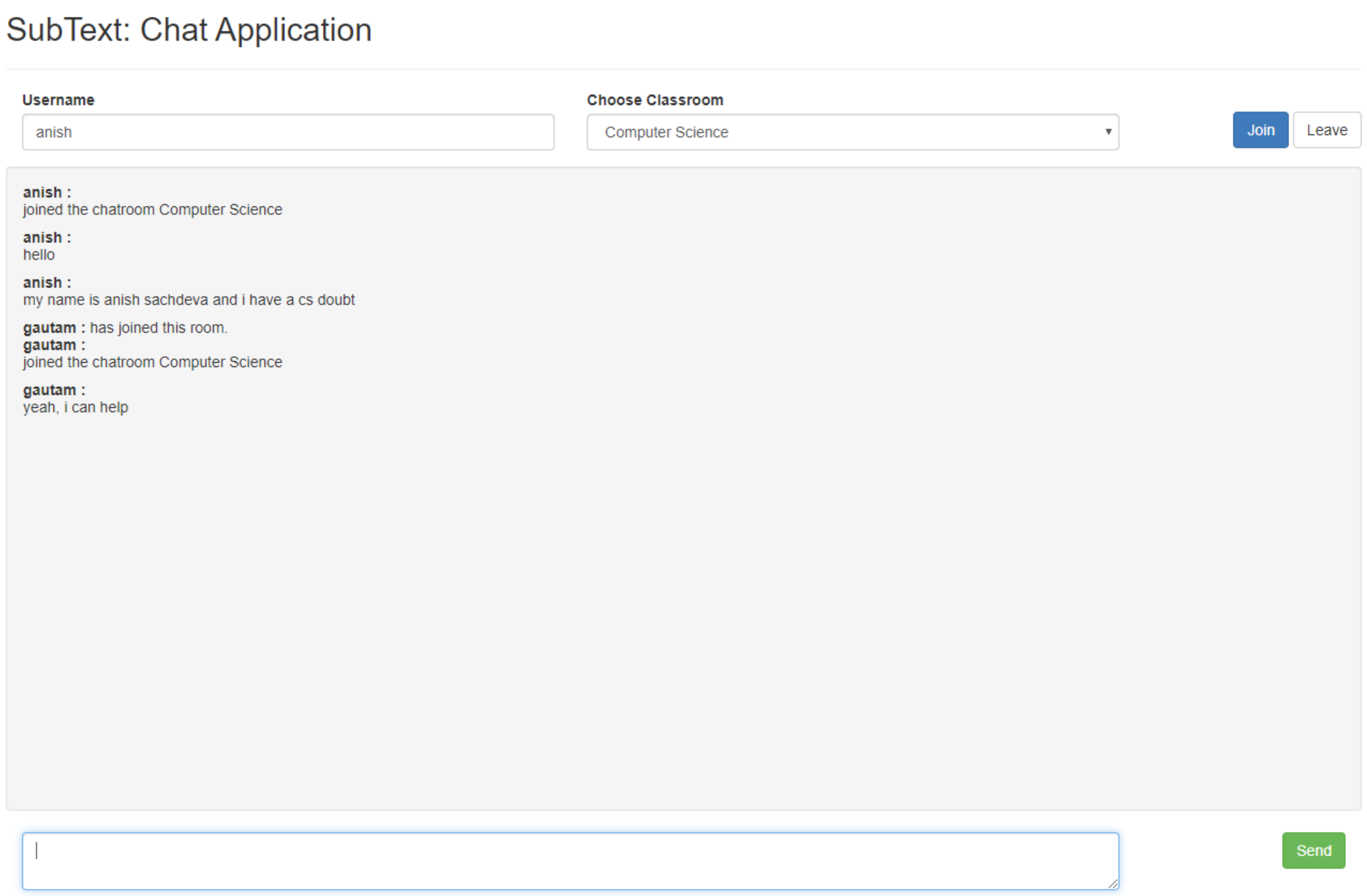
anish @ 15:18:56
okay sir, i had a doubt in 2 questions and will pass by on monday

gautam @ 15:19:55
i will also accompany you @anish 😃

Prototype II

- Added mark down support
- Added multiple classroom support
- Created markdown analyser and text parser from scratch

Prototype II: Text interaction



SubText: Chat Application

anish :
joined the chatroom Computer Science

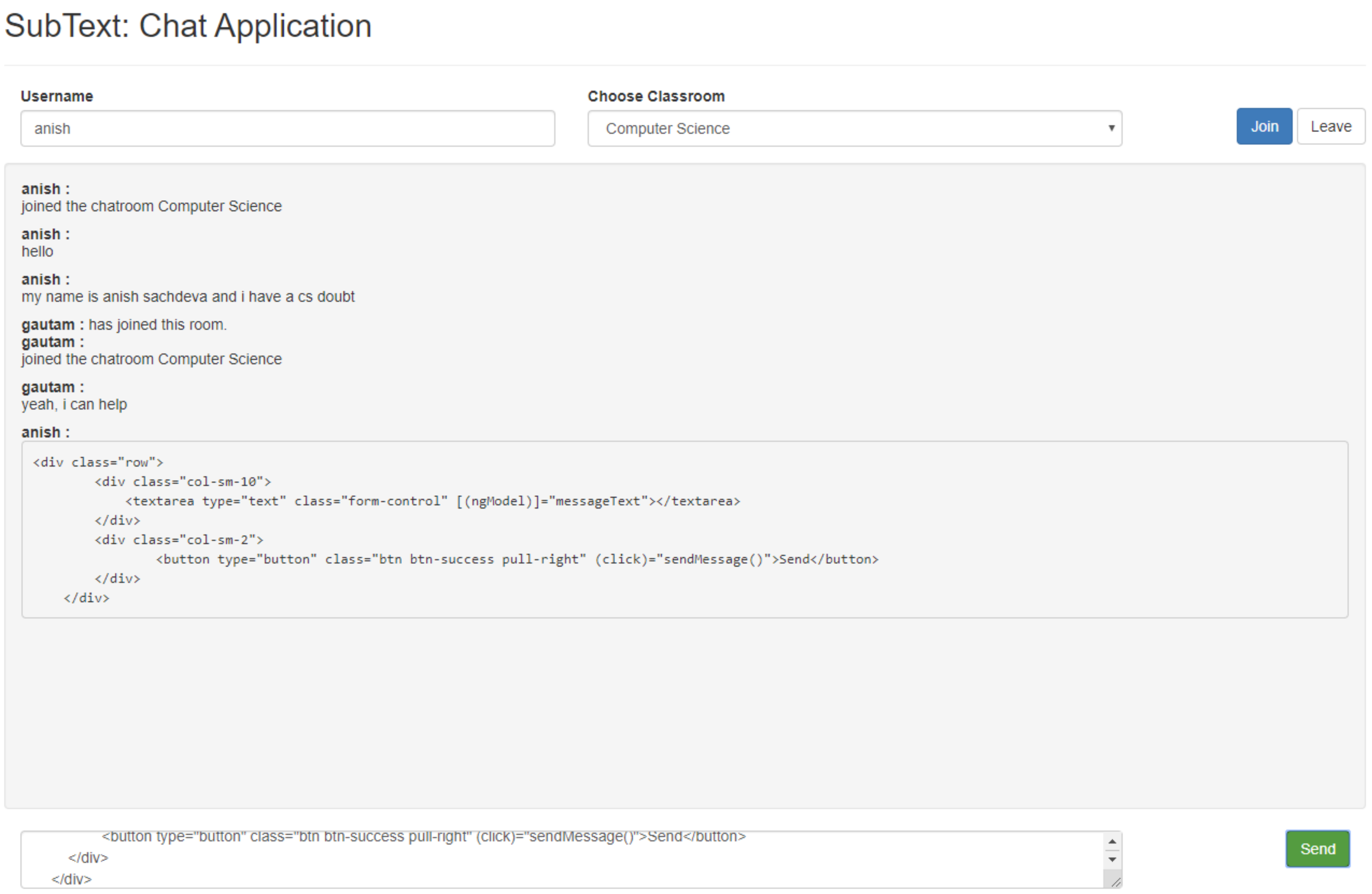
anish :
hello

anish :
my name is anish sachdeva and i have a cs doubt

gautam : has joined this room.
gautam :
joined the chatroom Computer Science

gautam :
yeah, i can help

Prototype II: Markdown + Text Interaction


SubText: Chat Application

anish :
joined the chatroom Computer Science

anish :
hello

anish :
my name is anish sachdeva and i have a cs doubt

gautam : has joined this room.
gautam :
joined the chatroom Computer Science

gautam :
yeah, i can help

anish :

```
<div class="row">
        <div class="col-sm-10">
            <textarea type="text" class="form-control" [(ngModel)]="messageText"></textarea>
        </div>
        <div class="col-sm-2">
                <button type="button" class="btn btn-success pull-right" (click)="sendMessage()">Send</button>
        </div>
    </div>
```

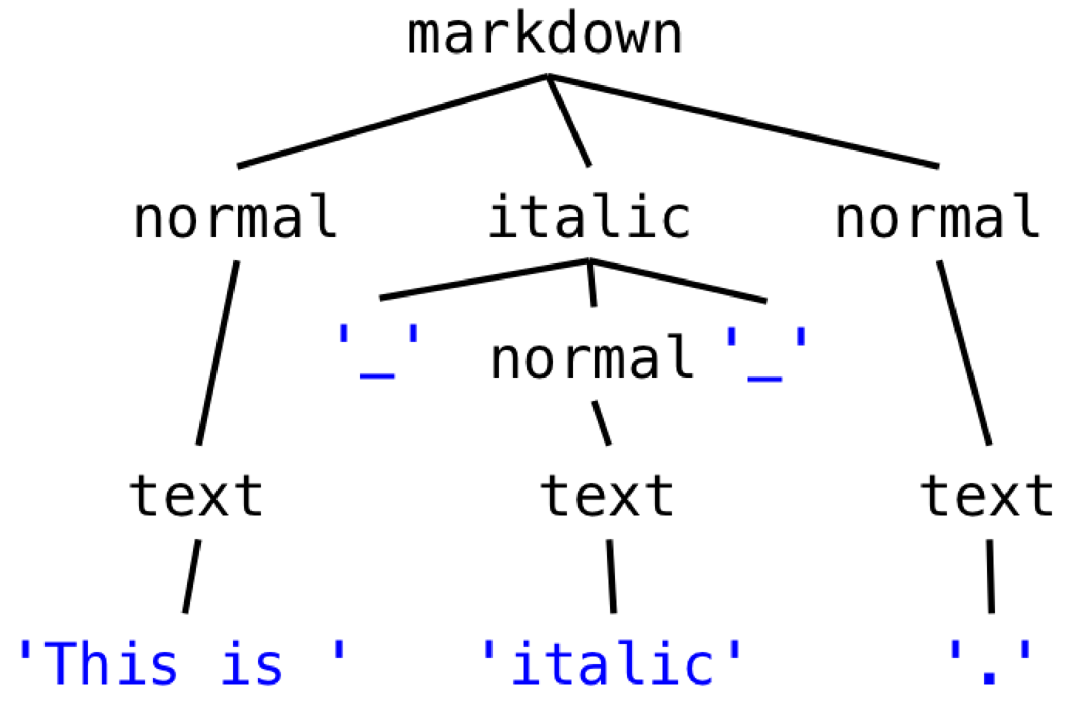
```
          <button type="button" class="btn btn-success pull-right" (click)="sendMessage()">Send</button>
      </div>
    </div>
```

Markdown

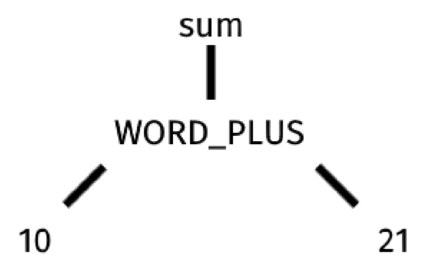
- **Markdown** is a lightweight markup language with plain text formatting syntax. Its design allows it to be converted to many output formats, but the original tool by the same name only supports HTML. Markdown is often used to format readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor.

- Developed by John Gruber and Aaron Swartz

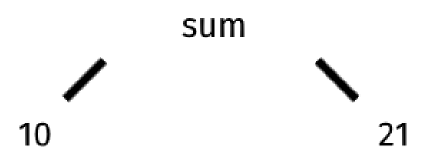
Lexical Parser



Parse Tree



Abstract Syntax Tree

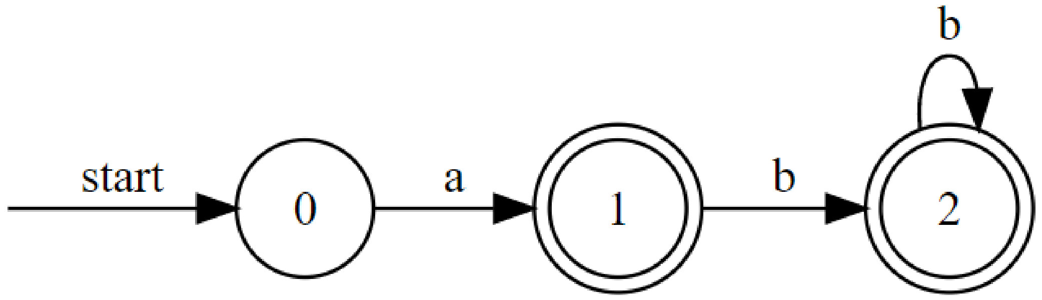


Finite Automata

A Finite Automaton is a 5-tuple ($Q, \Sigma, \delta, q_0, F$), where

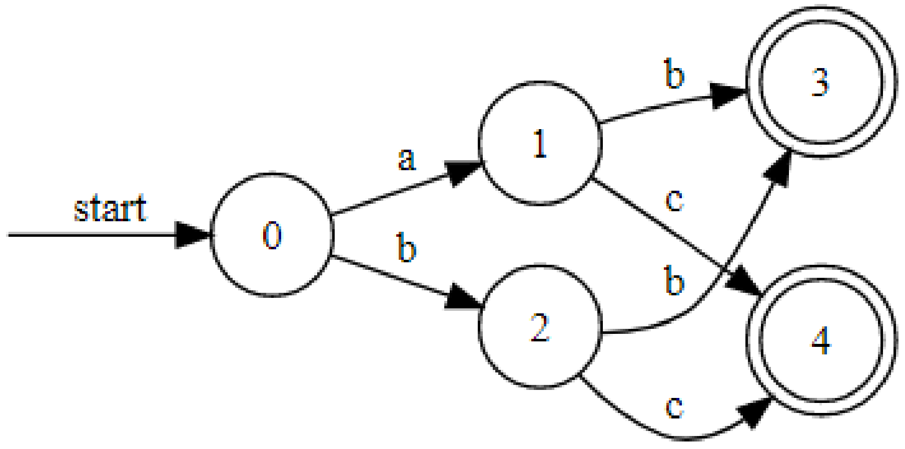
1. Q is a finite set called the **states**
2. Σ is a finite set called the **alphabet**
3. $\delta : Q \times \Sigma \to Q$ is the **transition function**
4. $q_0 \in Q$ is the **start state,** and
5. $F \subseteq Q$ is the **set of accepted states.**

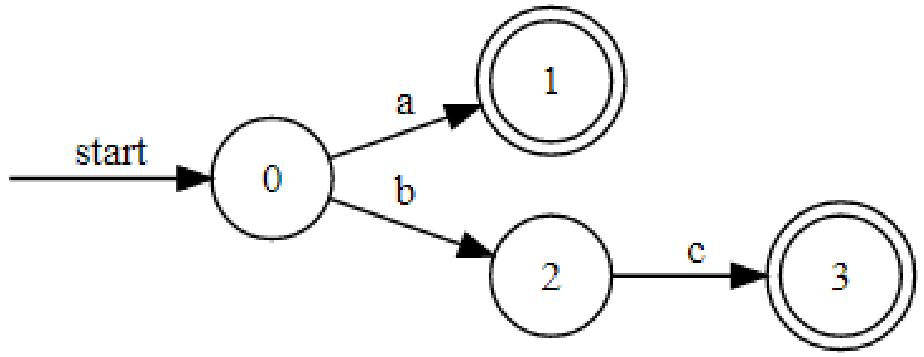
State	a	b
$\to 0$	1	Φ
1	Φ	2
2	Φ	2

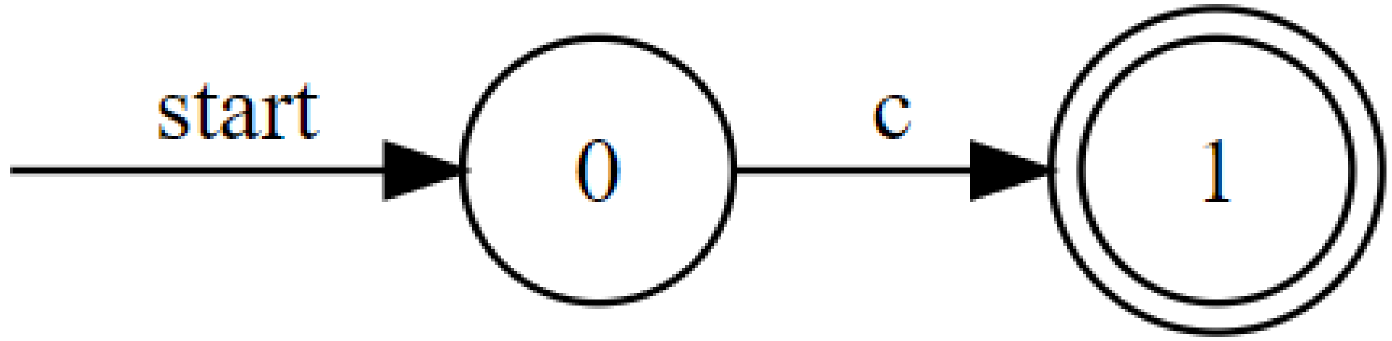


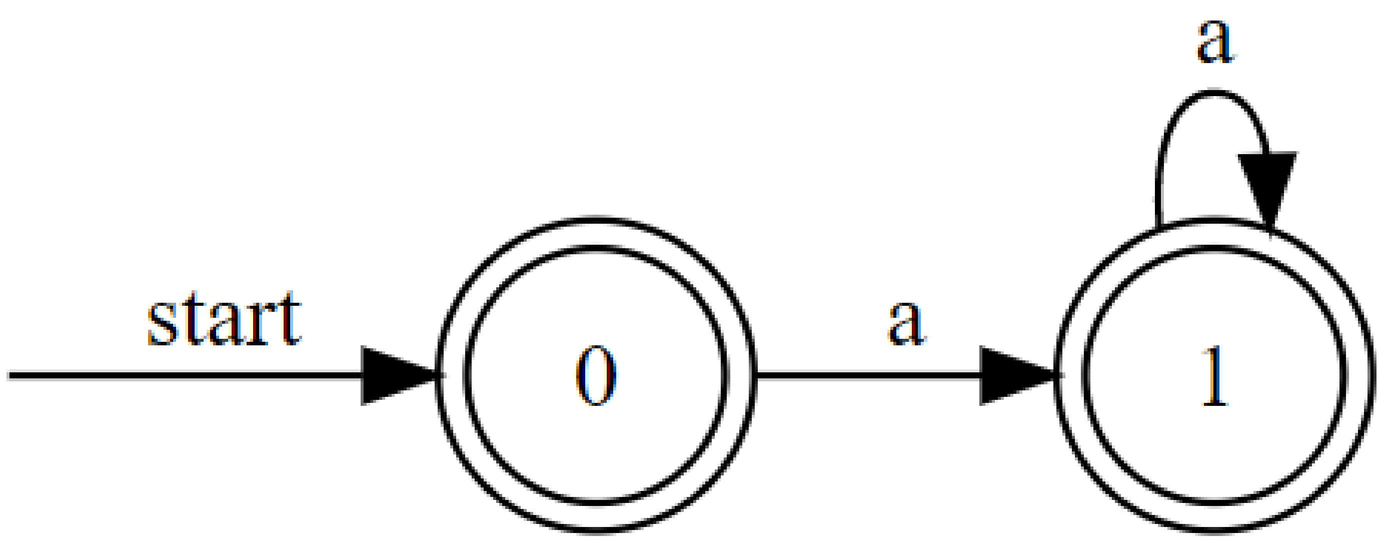
Finite State Automata Generator

- See this
- This is a finite state automata generation web application that builds deterministic and non-deterministic automata based on regular expressions.







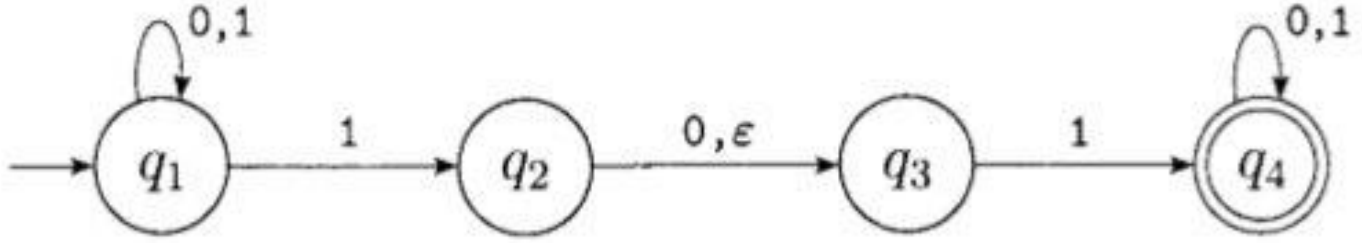


Deterministic Finite Automata (DFA)

Non Deterministic Finite State Automata (NDFA)

A Finite Automaton is a 5-tuple (Q, Σ, δ , $q_{0,}$ F), where

1. Q is a finite set called the **states**
2. Σ is a finite set called the **alphabet**
3. $\delta : Q \times \Sigma \rightarrow P(Q)$ is the **transition function**
4. $q_0 \in Q$ is the **start state,** and
5. $F \subseteq Q$ is the **set of accepted states.**



	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$,
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

Context Free Grammars

Formal Definition

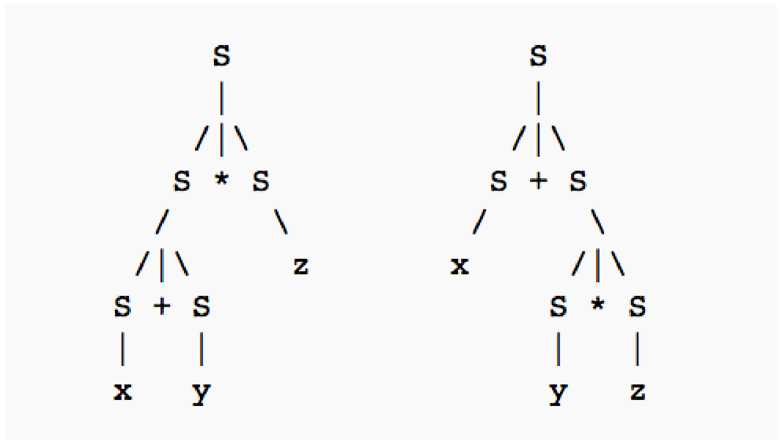
A context free grammar can be described by a 4 element tuple (V, Σ, R, S), where

V is a finite set of variables (which are non-terminal)

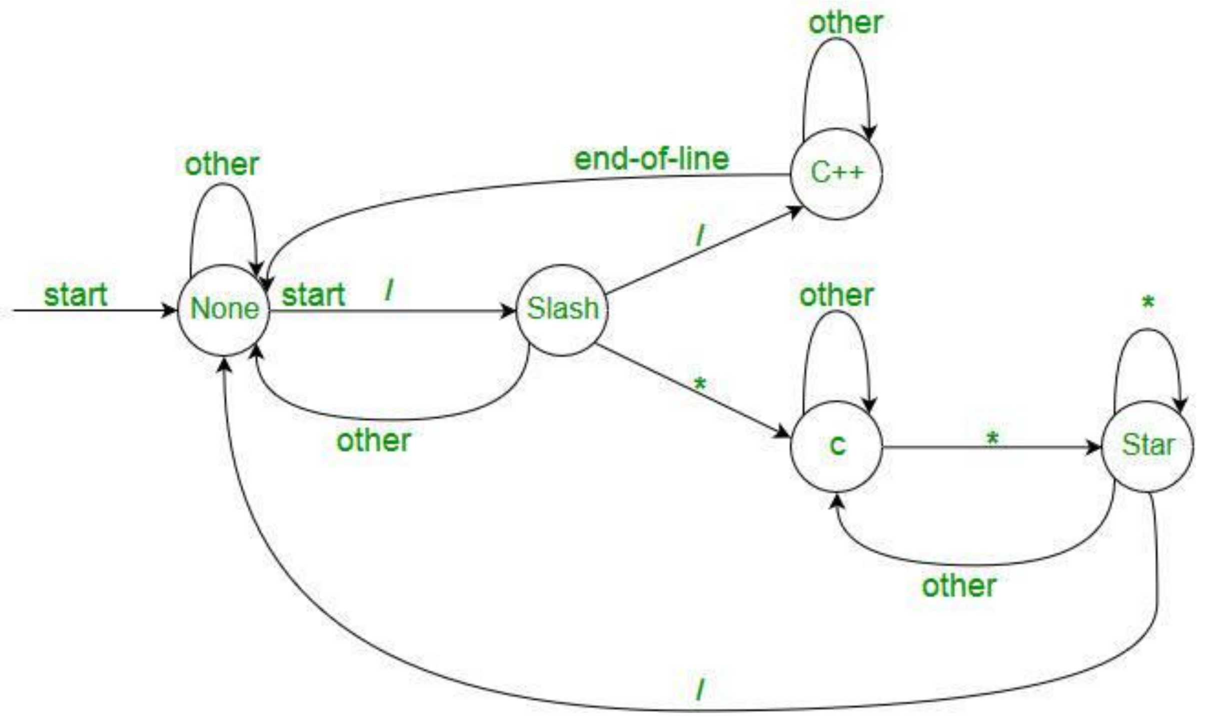
Σ is a finite set (disjoint from V) of terminal symbols

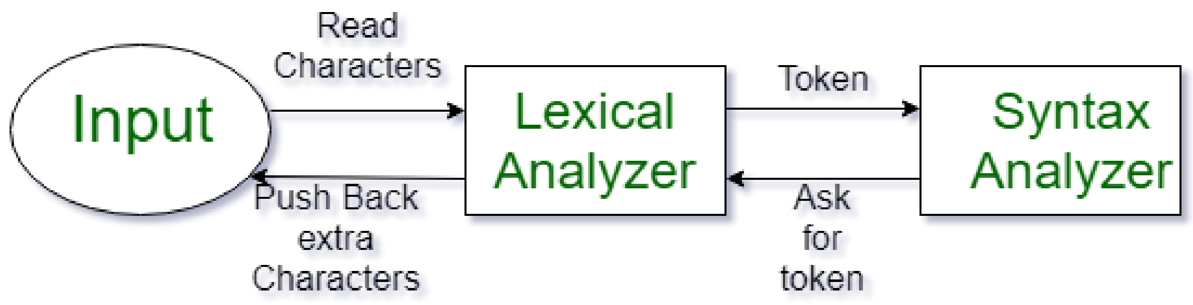
R is a set of production rules where each production rule maps a variable to a string $s \in (V \cup \Sigma)^*$

S (which is in V) which is a start symbol.



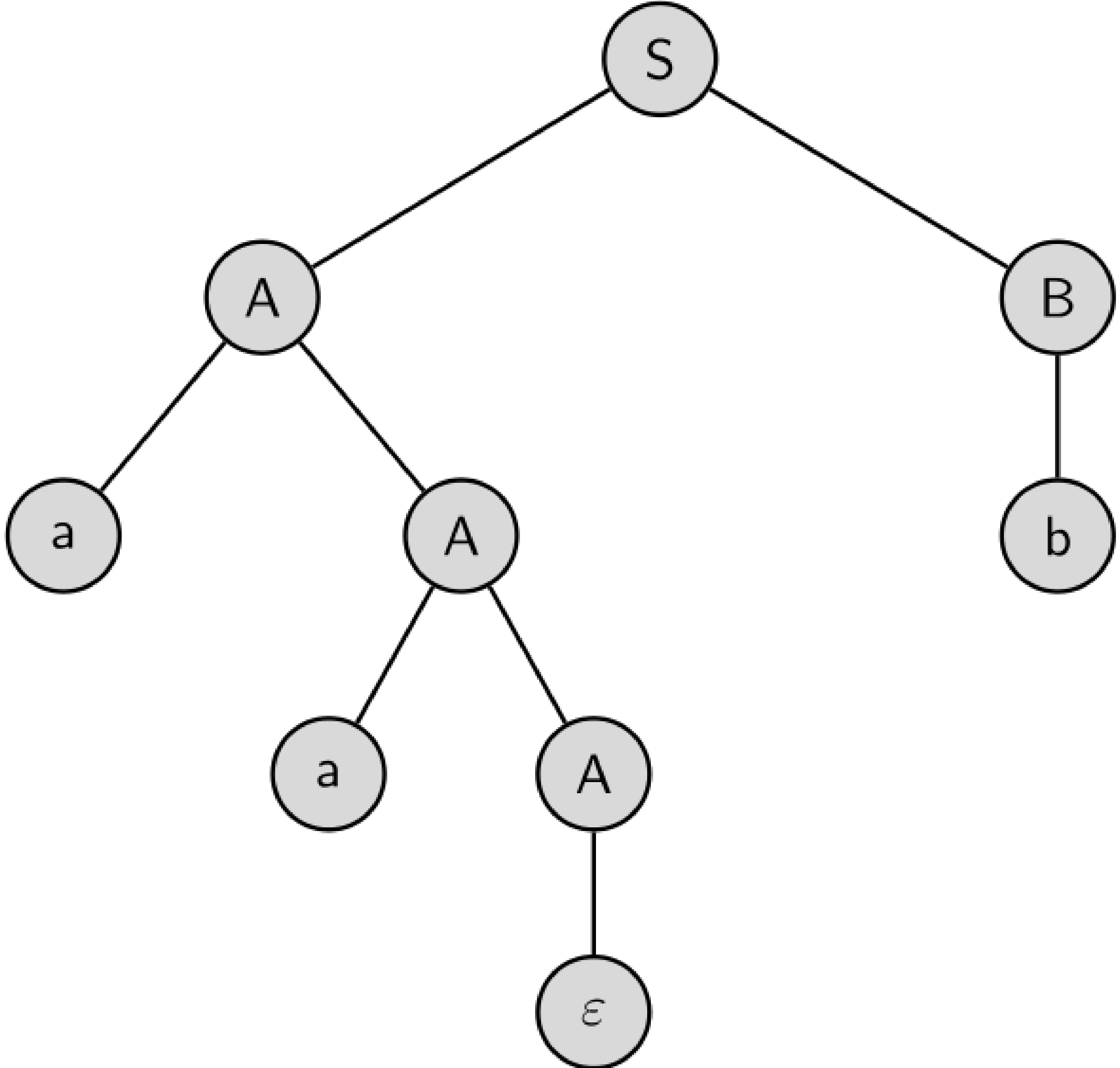
Lexical Parsing





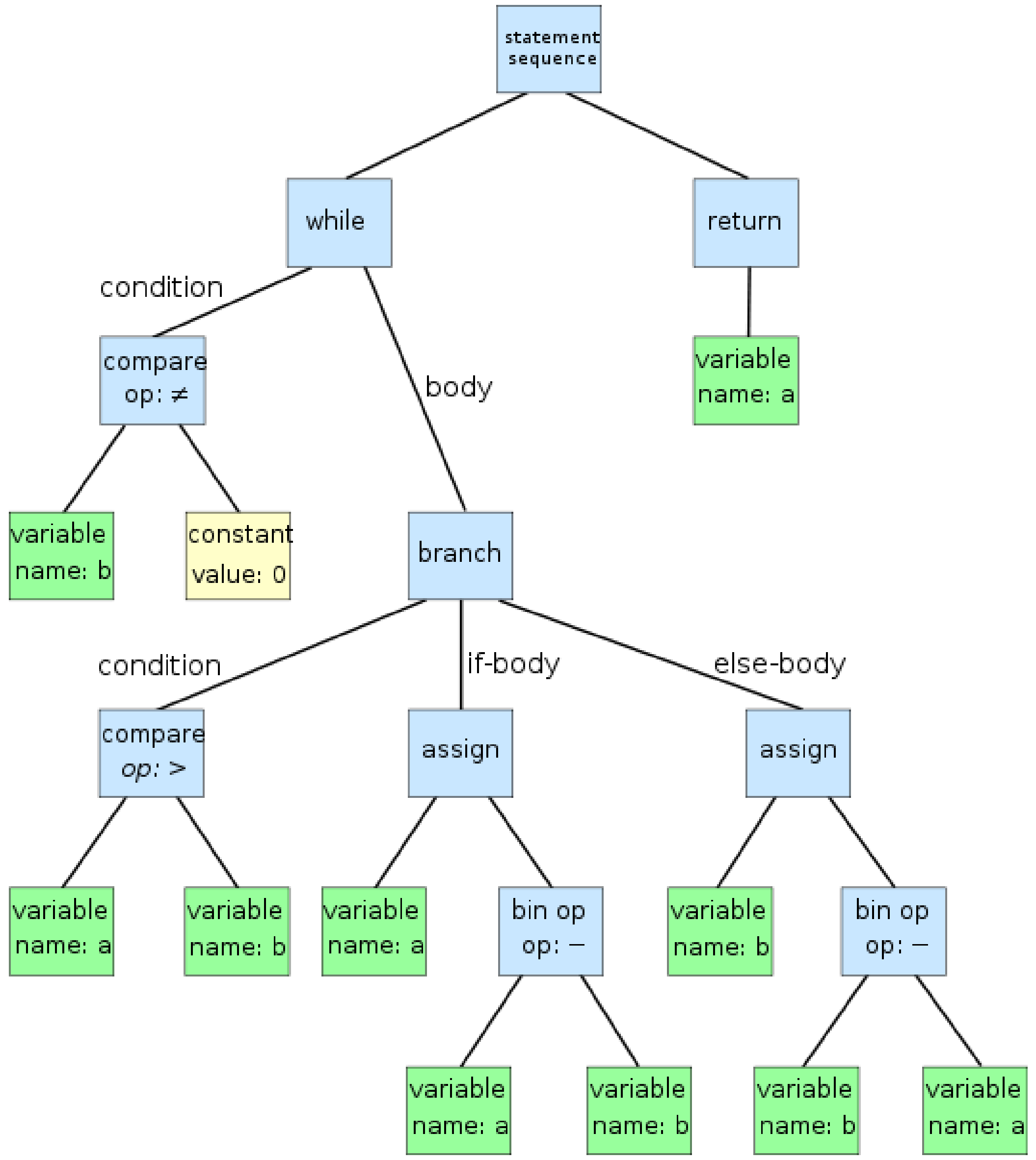
Parse Tree

- A **parse tree** or **parsing tree** or **derivation tree** or **concrete syntax tree** is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. The term *parse tree* itself is used primarily in computational linguistics; in theoretical syntax, the term *syntax tree* is more common.



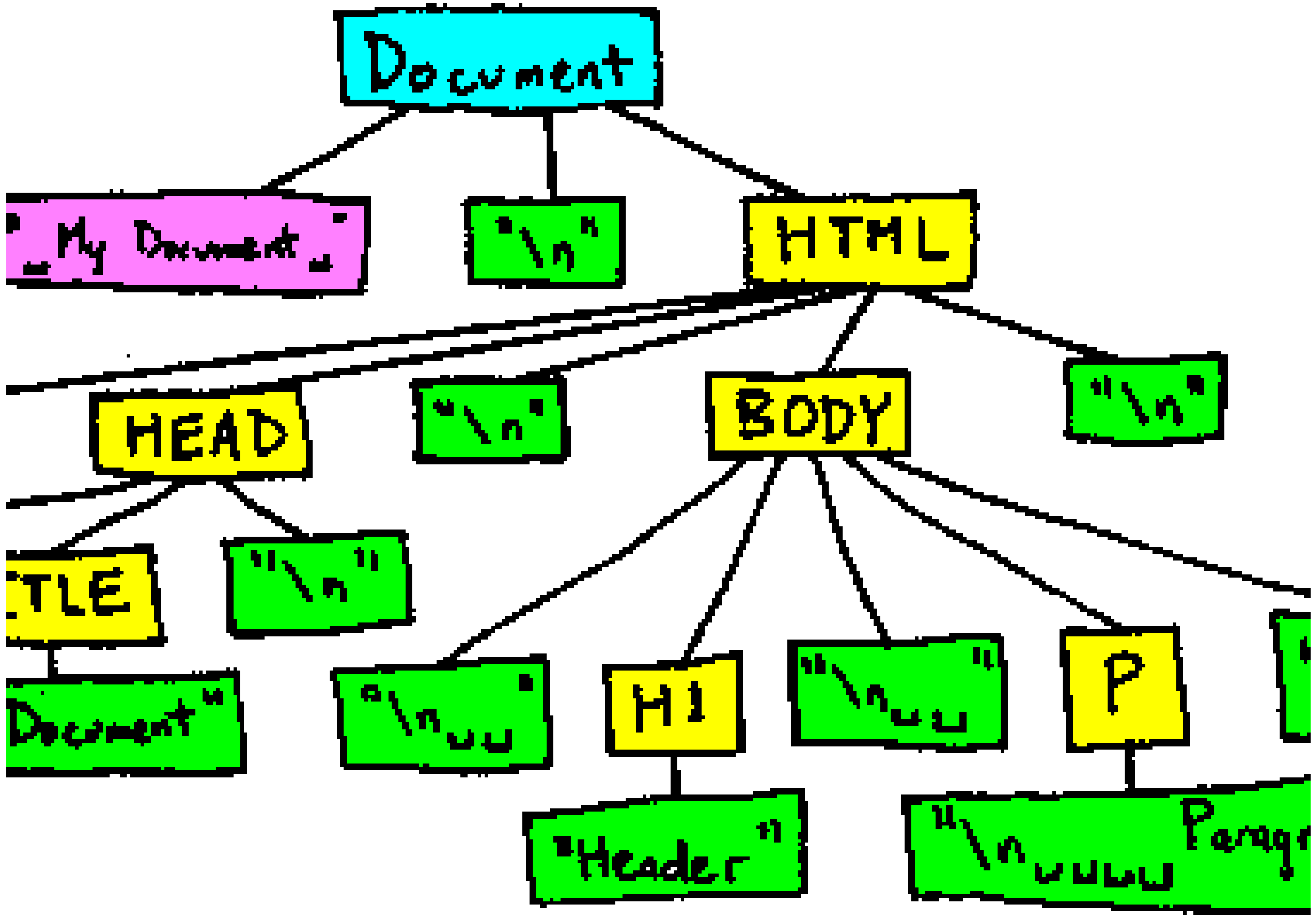
Abstract Syntax Tree

- In computer science, an **abstract syntax tree** (**AST**), or just **syntax tree**, is a tree representation of the abstract syntactic structure of source code written in a programming language. Each node of the tree denotes a construct occurring in the source code.



Markdown Implementation

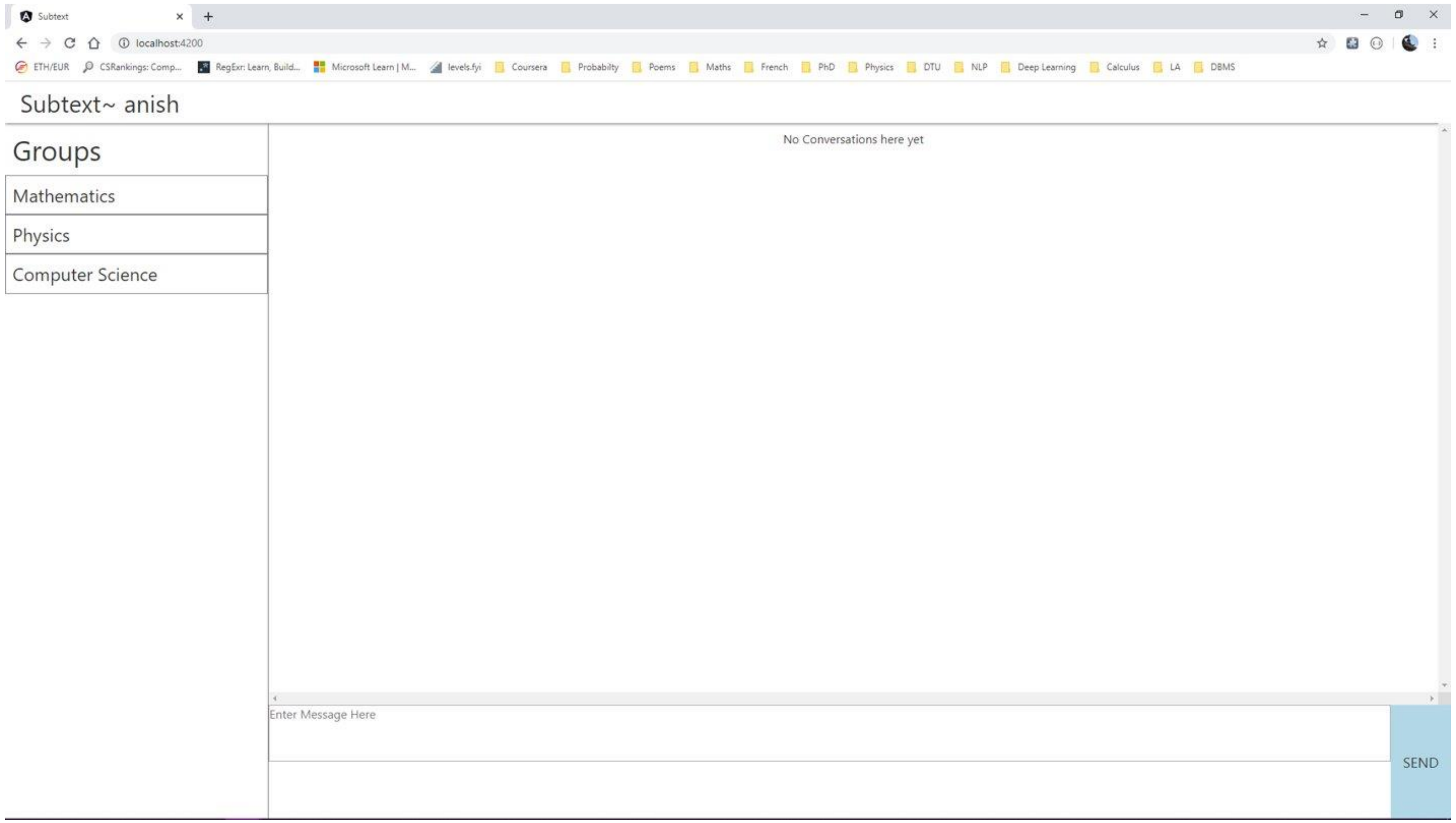
- The entire parser consists of many functions that each individually manage the successful conversion of a single markdown construct into an HTML node. A markdown construct here is referring to one single markdown formatting rule that applies e.g bold text or italic text. Here this atomic rule is referring to a single construct and I have created a singular function to handle and manage every such singular construct.



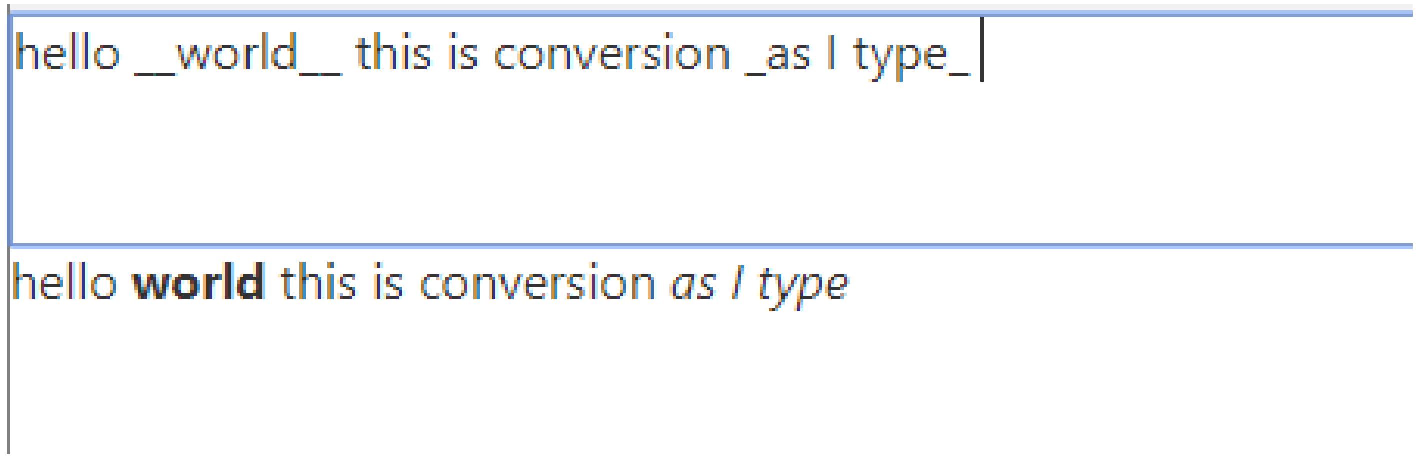
Prototype III

- Adds LaTeX Support
- Adds Mathematical fonts to represent Mathematical symbols
- Create an online compiler and parser that converts text to html based on LaTeX markup.
- This compiler works without any internet connection and is faster than OverLeaf!
- Also created a brand new interface for the application

Prototype III: Dedicated Groups Panel



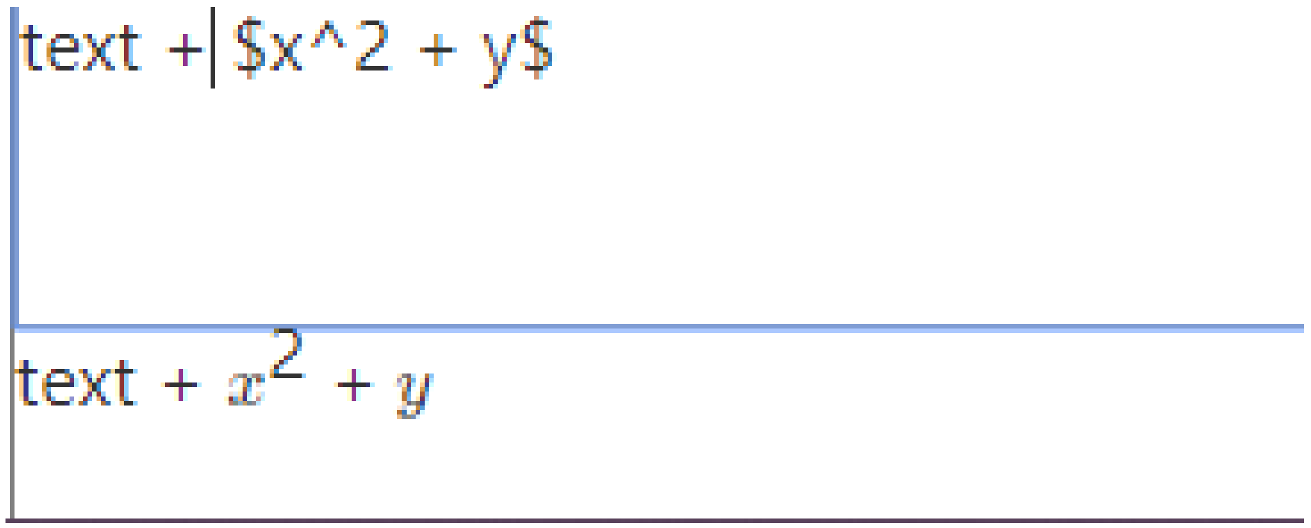
Prototype III: Hot-Reload Conversion of Markdown text into Markdown



hello __world__ this is conversion _as I type_

hello **world** this is conversion *as I type*

Prototype III: Conversion to LaTeX as we type

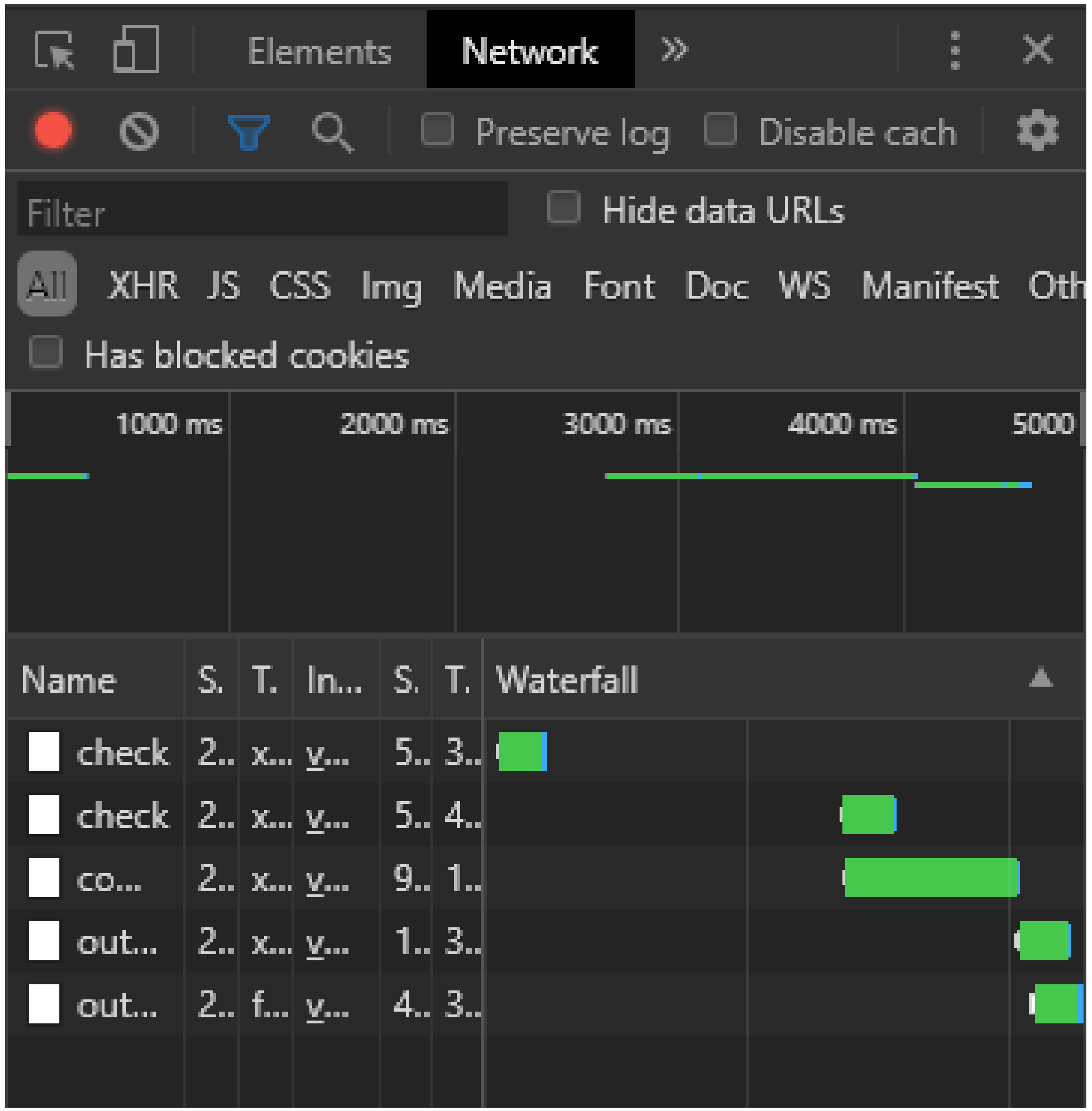


```
text + $x^2 + y$
```

text + $x^2 + y$

Overleaf vs. SubText LaTeX Conversion

- Open Overleaf Sample Project here
- SubText is 20x-400x faster!



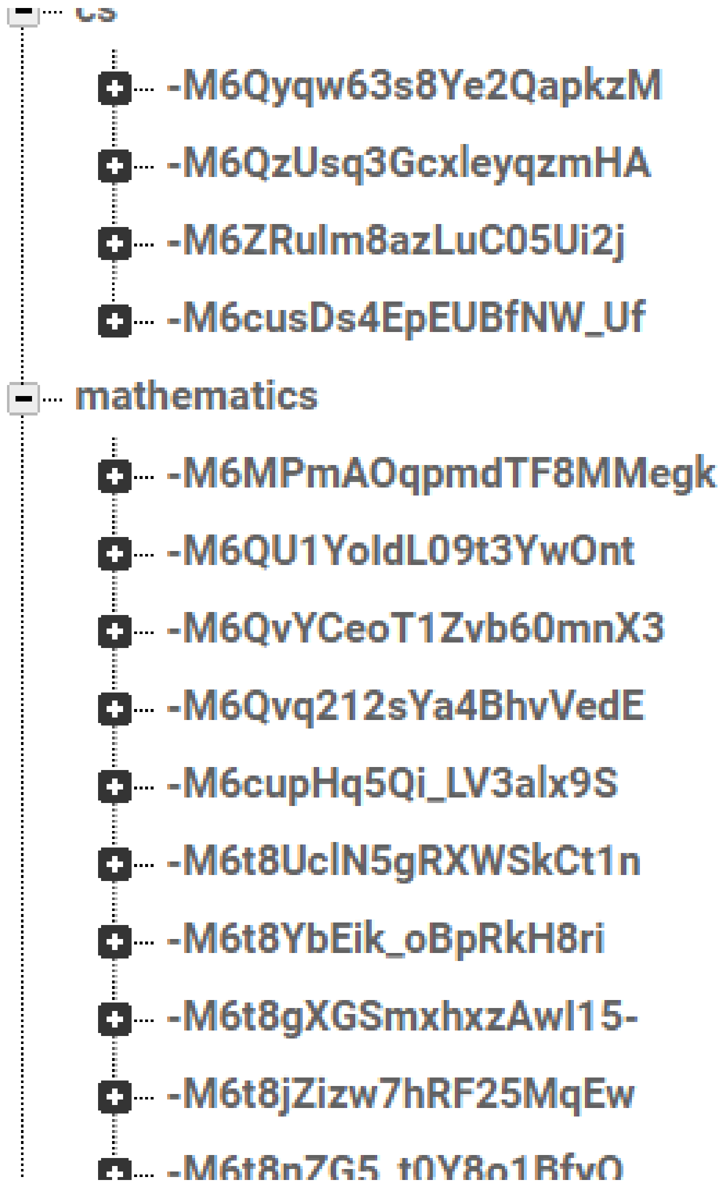
Name	S.	T.	In...	S.	T.	Waterfall
check	2..	x...	v...	5..	3..	
check	2..	x...	v...	5..	4..	
co...	2..	x...	v...	9..	1..	
out...	2..	x...	v...	1..	3..	
out...	2..	f...	v...	4..	3..	

Firebase Database

Firebase Database is a real time Database offered by Google.

It is very reactive and highly scalable.

Live data creation and modification can be seen here.



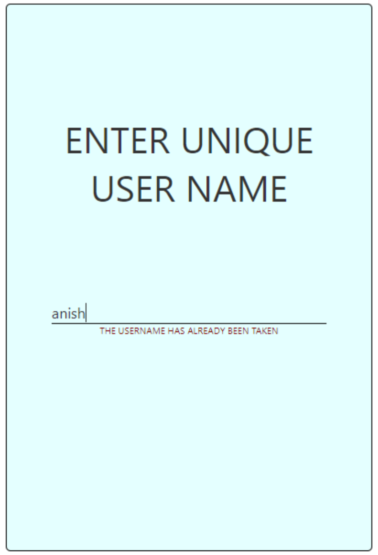
```
cs
    -M6Qyqw63s8Ye2QapkzM
    -M6QzUsq3GcxleyqzmHA
    -M6ZRulm8azLuC05Ui2j
    -M6cusDs4EpEUBfNW_Uf
mathematics
    -M6MPmAOqpmdTF8MMegk
    -M6QU1YoldL09t3YwOnt
    -M6QvYCeoT1Zvb60mnX3
    -M6Qvq212sYa4BhvVedE
    -M6cupHq5Qi_LV3alx9S
    -M6t8UclN5gRXWSkCt1n
    -M6t8YbEik_oBpRkH8ri
    -M6t8gXGSmxhxzAwl15-
    -M6t8jZizw7hRF25MqEw
    -M6t8nZG5_t0Y8o1BfvO
```

Persistent Message Store

See message database here.

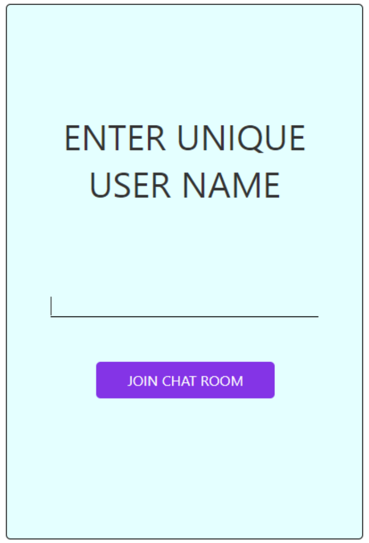
Multiple User Support + Login/Logout Screen

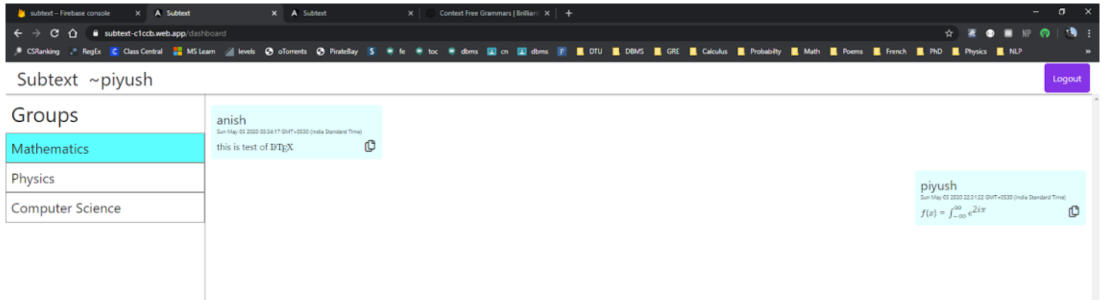
- See database live here.


ENTER UNIQUE USER NAME

anish

THE USERNAME HAS ALREADY BEEN TAKEN


ENTER UNIQUE USER NAME



Subtext ~piyush

Groups

Mathematics

Physics

Computer Science

anish
Sun May 03 2020 03:34:17 GMT+0530 (India Standard Time)
this is test of LaTeX

piyush
Sun May 03 2020 22:31:22 GMT+0530 (India Standard Time)
$f(x) = \int_{-\infty}^{\infty} e^{2i\pi}$

Future Scope – The Casual User

PERSISTEN USER INFORMATION ON THE DESKTOP USING COOKIES

BUYING A DOMAIN NAME LIKE SUBTEXT.COM OR SUBTEXT.CHAT

GROUP CREATION AND CHAT CREATION FUNCTIONALITY

ADDING A ML/AI KEYBOARD AGENT THAT DETECTS AUTOMATICALLY WHETHER A PERSON WISHES TO TYPE MARKDOWN OR LATEX AND THEN AUTOMATICALLY SWITXCHING TO THAT CONTEXT

ADDING OTHER CONTEXTS SUCH AS PICTURES, LIVE CODE SNIPPETS, LISTS, QUIZES, CALNDER EVENTS ETC.

ADDING EMAIL AND OAUTH BASED PERSISTENT USER AND MESSAGE STORAGE

Future Scope – Businesses & Organizations

Future Scope - Organizations

Adding organizational capabilities and allowing organizations to create custom domain and create multiple internal teams with many groups.

An organization can register its individual name like *{abc}* and then access the domain like *abc.subtext.chat/group-name*

Inside a group, group owners will have the ability to create multiple channels for discussing and communicating on various issues and they will be able to do so individually in each channel using Markdown, LaTeX and all the other contexts that te application can automatically infer.

Future Scope - Organizations

For communicating inside a particular channel they can use *{organization-name}.subtext.com/{group}/{channel}*

And for chatting with an individual user they can use *{organization-name}.subtext.com/user/{user-name}*

Organizations - Revenue Model

- Will be charged on a per user basis so larger organizations will be billed more as compared to smaller organizations.
- The features required to make this production ready are feasible and minimum funding is required and with the support of DTU and the startup incubator and an additional good programmer it is very likely that the world's most powerful communication application may come out of DTU.

Conclusion

Conclusion

Users can express themselves better when they do not have to think what kind of messages they can or can't send.

They are able to better express themselves and additional markup languages such as MarkDown and LaTeX draws professionals from many different backgrounds to make this application much more attractive.

Additional features such as Lists, Quizes and Calender events will further make this application lucrative to non-technical and non-industry specific persons.

All parsers and lexical analyzers are implimeted on the client side code meaning that all transformations and rendering happens without calling the server and without need of internet.

Any Additional Questions?

Important Links

Subtext Repository

Subtext Deployed

Subtext Database [Firebase]

Author's LinkedIn Profile

Author's GitHub Profile

Thank You :)