

Project Thor

Team Members:

Adonay Pichardo

Jared Blanco

Josh Temel

Luke Boneburger

Faculty Advisor:

Dr. Sid Bhattacharyya

Client:

Dr. Amitabh Nag

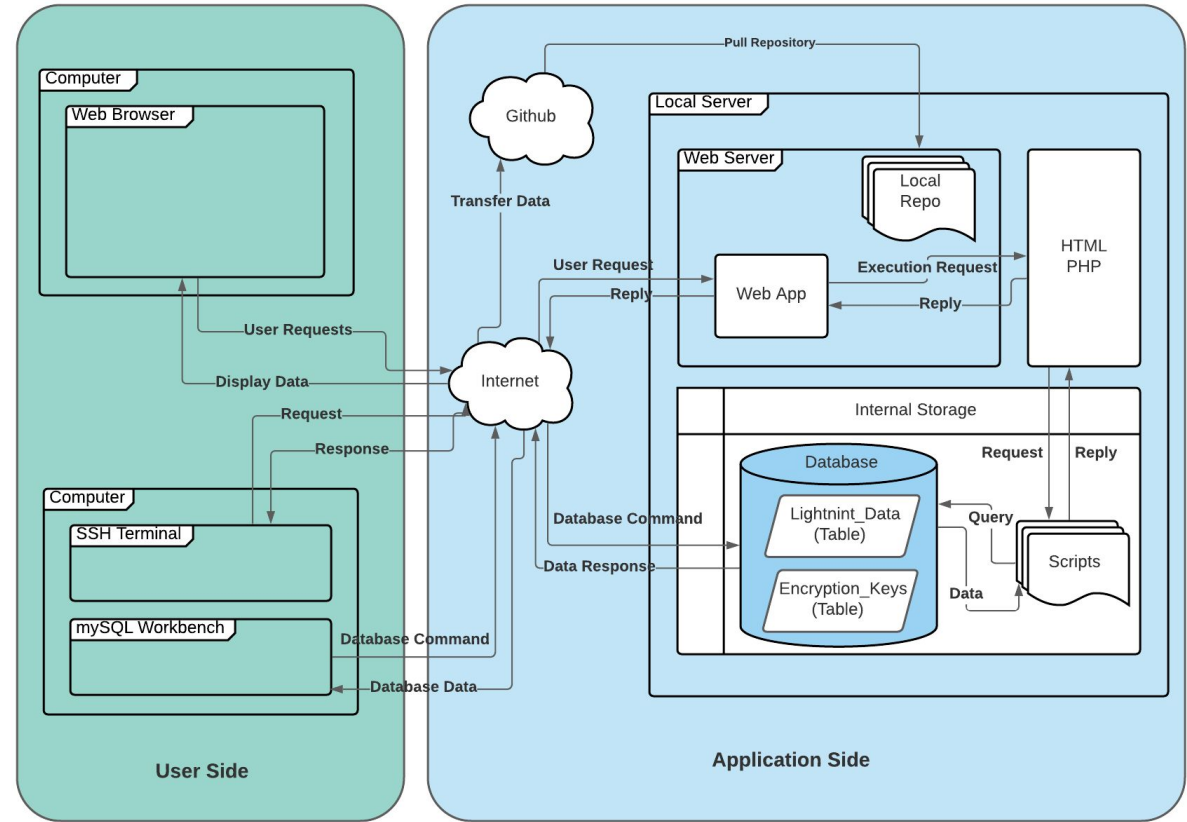
[Google Slides](#)

Milestone 2 Task Matrix (Nov 1)

Task	Adonay	Jared	Josh	Luke
1.Update "Hello World" demos	Update data set demo 25%	Update data set demo 25%	Update database demo 25%	Update web application demo 25%
2. Update documentation	Read and review edits	Read and review edits	Update	Read and review edits
3. Setup school provided server environment	Work with Josh to set up server 50%	None	Work with dr sid to obtain hardware. Work with Adonay to set up 50%	None
4. Automated data transfer script creation	Offer help / troubleshoot	Create and implement	Offer help / troubleshoot	Offer help / troubleshoot
5. Design 25% of website layout (excluding content)	Read and review	Read and review	Work on 50%	Work on 50%
6. Show entropy findings	Work on 50%	Work on 50%	Offer help / troubleshoot	Offer help / troubleshoot
7. Display basic generated key from database to website (finish MD5 implementation)	Work on calculation 40%	Work on calculation 40%	Offer help / troubleshoot	Work on website implementation 20%

Changes to Project Documentation

- Design Doc -- Updated System Architecture Diagram



Demos

1. [Live Web Application](#)

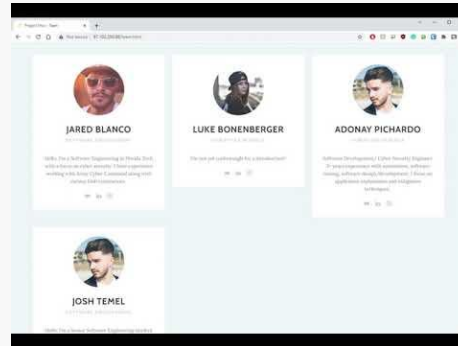
2. Automated Data Transfer

3. Current Data Entropy

3. MD5 Hash Generation

Demo 1: Live Web Application

1. Live Web Application



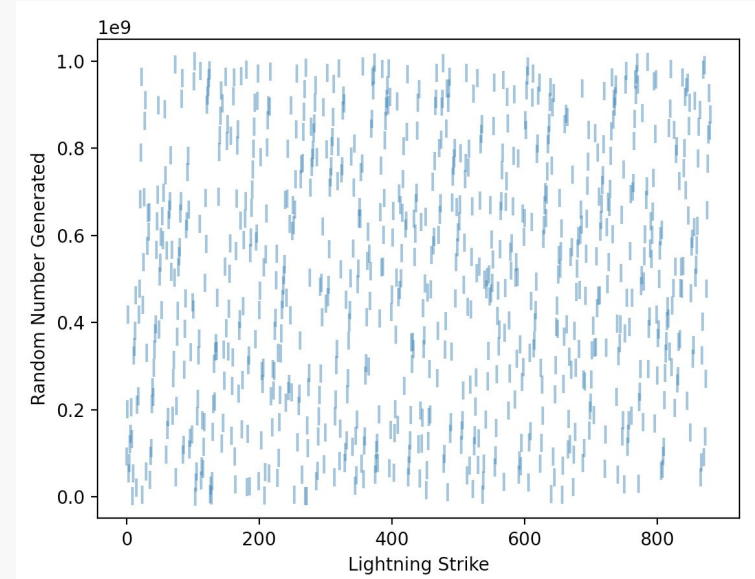
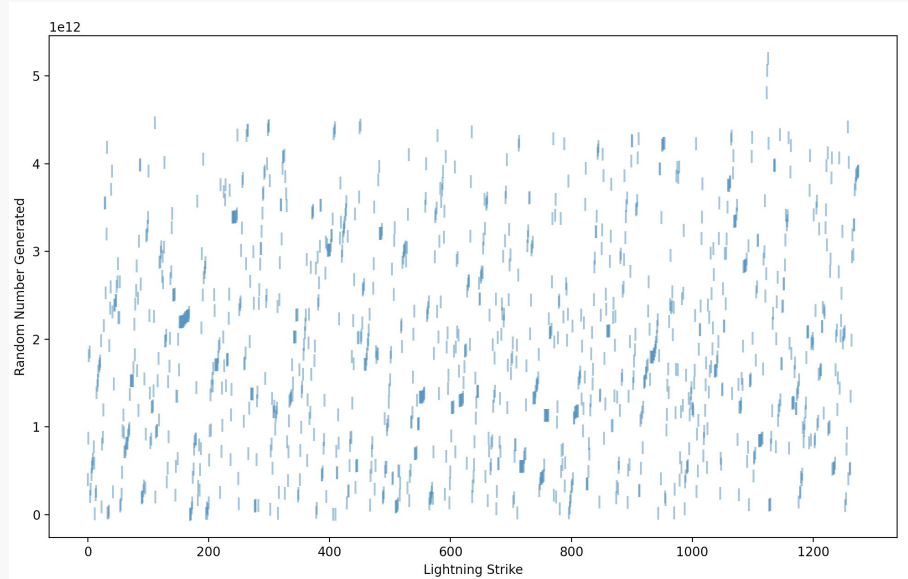
Demo 1 video

Demo 2: Automated Data Transfer

Video Demo

```
~/Documents/FIoTech_Fall_2021/thor-repo/src/python_scripts — -bash
Adonays-MacBook-Pro:python_scripts adonay$ python3 getDatabase.py credentials Lightning_Data lightning_record
lightning_record
strike_time lat lon rise_time fall peak_cur
2021-8-7 2:0:0.121214720 42.7154 -88.6102 3.1 17.1 4.0
2021-8-7 2:0:0.133076736 43.4163 -101.5432 1.8 3.1 6.0
2021-8-7 2:0:0.144075800 42.7348 -88.5178 3.6 2.1 9.0
2021-8-7 2:0:0.154391040 42.7354 -88.517 3.2 2.1 7.0
2021-8-7 2:0:0.167083520 42.7348 -88.5189 4.6 2.7 10.0
2021-8-7 2:0:0.200858624 42.8698 -101.0557 7.2 27.9 -25.0
2021-8-7 2:0:0.204898560 42.7331 -88.5228 3.6 5.7 -4.0
2021-8-7 2:0:0.211297280 42.8774 -89.0291 12.4 5.2 8.0
2021-8-7 2:0:0.21871616 40.0922 -101.5134 2.1 6.8 3.0
2021-8-7 2:0:0.231087616 42.883 -89.0828 2.2 4.2 4.0
2021-8-7 2:0:0.2416896 40.8078 -104.895 3.5 4.0 5.0
2021-8-7 2:0:0.324540160 40.8375 -104.9014 1.2 1.4 -4.0
2021-8-7 2:0:0.343484160 40.865 -104.8834 7.5 17.9 -4.0
2021-8-7 2:0:0.357552640 40.7648 -105.0718 6.2 1.4 -3.0
2021-8-7 2:0:0.381311488 40.8541 -104.879 13.1 15.7 -7.0
2021-8-7 2:0:0.393331584 40.8266 -104.9274 16.9 4.3 -8.0
2021-8-7 2:0:0.401404160 40.733 -105.0937 17.6 3.0 5.0
2021-8-7 2:0:0.408315392 40.827 -104.9268 1.1 1.6 -3.0
2021-8-7 2:0:0.418180096 42.8695 -101.056 4.0 32.2 -21.0
2021-8-7 2:0:0.418247680 26.3293 -108.484 27.8 14.5 -14.0
2021-8-7 2:0:0.427521024 42.8435 -101.1153 6.7 4.2 10.0
2021-8-7 2:0:0.463367680 40.7509 -104.9381 1.1 1.6 3.0
2021-8-7 2:0:0.477646336 44.2837 -101.9117 19.0 2.5 -5.0
2021-8-7 2:0:0.53221888 40.8182 -104.9674 1.2 2.3 3.0
2021-8-7 2:0:0.54703360 42.7332 -88.5186 3.2 2.0 5.0
2021-8-7 2:0:0.61746944 40.8383 -104.9302 1.1 6.0 4.0
2021-8-7 2:0:0.682120192 48.8573 -106.5649 2.5 3.2 6.0
2021-8-7 2:0:0.682663072 40.8471 -106.5595 11.4 9.6 7.0
2021-8-7 2:0:0.697864448 32.3558 -112.0508 9.9 8.2 8.0
2021-8-7 2:0:0.77378048 42.7352 -88.5182 8.2 2.1 7.0
2021-8-7 2:0:0.789153024 39.5853 -102.3431 8.0 9.2 -4.0
2021-8-7 2:0:0.92156416 42.8025 -101.1417 5.1 6.2 -9.0
2021-8-7 2:0:0.963484416 42.9272 -101.1993 17.8 13.6 -3.0
2021-8-7 2:0:1.10459904 40.1923 -101.7042 1.5 8.0 -8.0
2021-8-7 2:0:1.440847616 43.2226 -101.1822 1.2 3.8 -3.0
2021-8-7 2:0:1.4867072 41.312 -100.8744 5.0 10.6 -2.0
2021-8-7 2:0:1.538110976 43.1735 -101.2108 10.0 8.0 12.0
2021-8-7 2:0:1.5977600 42.9101 -89.1551 12.3 2.3 7.0
2021-8-7 2:0:1.687106384 42.6837 -88.4766 4.2 4.7 7.0
2021-8-7 2:0:1.861891072 42.6400 -100.8611 23.5 21.9 16.0
2021-8-7 2:0:1.989772288 42.6438 -100.8622 15.8 13.1 14.0
2021-8-7 2:0:1.910238976 39.3677 -64.5349 9.8 24.1 -83.0
2021-8-7 2:0:1.122758912 25.9661 -79.1781 2.1 3.5 9.0
2021-8-7 2:0:1.169384704 40.473 -105.3014 1.7 23.4 -2.0
2021-8-7 2:0:1.265914112 42.8877 -89.1758 9.7 5.0 -5.0
2021-8-7 2:0:1.337994752 40.4085 -105.3585 3.0 10.7 4.0
2021-8-7 2:0:1.386566144 43.3488 -101.2676 11.2 4.0 0.0
2021-8-7 2:0:1.431242752 40.1401 -101.5447 9.8 4.6 -5.0
2021-8-7 2:0:1.433896960 40.1389 -101.5368 0.0 0.0 -3.0
2021-8-7 2:0:1.434003968 40.1399 -101.5399 1.3 4.7 -4.0
2021-8-7 2:0:1.512482560 43.0649 -101.3742 16.2 3.9 4.0
```

Demo 3- Milestone 1 vs Milestone 2



Demo 3: Data Analysis

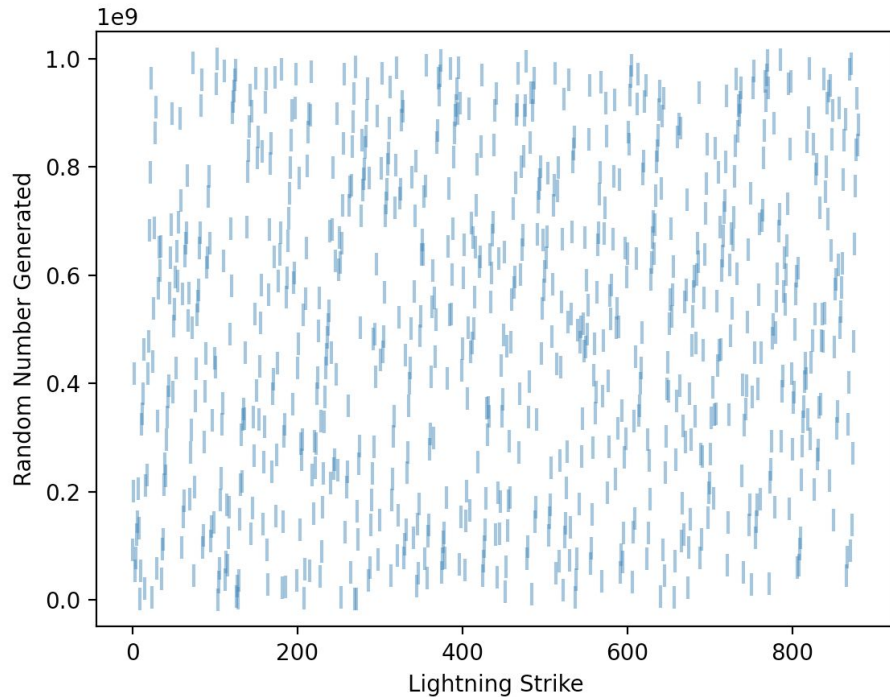
Problems addressed:

- Low variability in data
- Longitude and Latitude trends
- Data clumps

	Latitude	Longitude	TestLatitude	TestLongitude
0	42.8025	101.1417	80250.0	14170.0
1	42.8698	101.0557	86980.0	5570.0
2	42.8695	101.0560	86950.0	5600.0
3	42.8435	101.1153	84350.0	11530.0
	EllipseAngle	TestEllipseAngle	Nanosecond	
0	131.99	13199.0	92156416	
1	100.44	10044.0	200858624	
2	103.37	10337.0	418180096	
3	82.17	8217.0	427521024	

Latitude	StdDev = 6.77
Longitude	StdDev = 10.22
TestLatitude	StdDev = 29184.63
TestLongitude	StdDev = 28315.23
EllipseAngle	StdDev = 49.32
TestEllipseAngle	StdDev = 4931.93
Nanosecond	StdDev = 292230612.59

Demo 3- Number Generated vs Strike Number



```
df['Range'] = np.transpose(range(0, len(df['Latitude'])))
df['TestLatitude'] = (df['Latitude'] % 1) * 100000
df['TestLongitude'] = (df['Longitude'] % 1) * 100000
df['TestEllipseAngle'] = df['EllipseAngle'] * 100
```

```
Version2 = (df['TestLatitude'] + df['TestLongitude'] +
            df['Nanosecond'] + (df['EllipseAngle'] * 100))

averageDifference = averageDiff(Version2)
Version3 = []
prevNumber = 0
# version3: excluding any num within (10% of average difference)
for number in Version2:
    if abs(number - prevNumber) > (averageDifference / 10):
        Version3.append(number)
    prevNumber = number
```

Demo 4- MD5 Generation

```
#####
# Checks if getDatabase.py is being used correctly
#####
def checkCLI(argv):
    if len(argv) < 1:
        stdout.write(f'USAGE: > python3 {argv[0]} <some string to hash>\n')
        stdout.write(f'REQUIRED - a string to create a hash.\n')
        exit()

#####
# Checks if MD5hash.py is being used correctly
#####
def createMD5hash(key):
    from hashlib import md5
    return md5(key.encode("utf-8")).hexdigest()
# <Functions End>

def main():
    #####
    # Verify the program has been called correctly
    #####
    checkCLI(argv)

    #####
    # Create the MD5 hash for every key given
    #####
    while (1):
        key = stdin.readline().rstrip()
        if len(key) == 0:
            break
        stdout.write(f'{createMD5hash(key)}\n')

if __name__ == "__main__":
    main()
```

Video Demo

```
~/Documents/FIoTech_Fall_2021/thor-repo/src/python_scripts --- -bash
Adonay-MacBook-Pro:python_scripts adonay$ python3 getDatabase.py credentials Lightning_Data lightning_record | python3 MD5hash.py
10037d1a125353c597d6df89150b45
c17263ff0f0cde73e2609dae733a579
ec8a2c7cab6286dcabdccefa8a950a5b
ec15ac0b4ba30f6553ae4348f8b1999d
0c6337da5bf6639c3ad77abefb56879c
92e1e5b2fb1174ba73529d687044382
0f6779f9143f34ab060a0f5dc546936
8dd089a4ea9239432804df5c78736cb
849275b7695f40a38fc80e942fba0e7
a70dea5cc5f037b3e63c7f2fc9585b44
ca9d0f00c58beb1ccd1ed9aa95dde5ec
9398c0b89d7ce238c835386b32625deb
94d7c0bba1c1c2f87017b627b05f89c2
a7036b7fd675a93d38cdcd2b33a0a071
7020d9ba6614ad8858e7e34a43b69298
e7f914b51b18f8acf94dd7f31b4e4e4e
6ae941cd859cd7d6c53c143332ca9ecf
1a71325bf566a43e8d19ac9f06e33b98
c4e04f4b136ae92970d1e5a08e9936c
ad6c2aff1f6863f92d12f473c2a2968b
eb0e0855713e91f5486d852066a7164
dc9478d65d8052cfadb870bad78c77e
5cef582b5262ad4205a5c3e93cddcbd
d09c39d767c0cdad5733ee35152b16
801360d51a1c08973f507e5a0de753
5f16aaa156d1c9f1efca08f8c26d992
4f8cbfce0f93d0a16efd133e599ead8c
2d8f1fa84665f3eba9b71bcf4a64ae0
04654b04ea704b6f87db3f963904cf8
fdcef30981126713c48288ee3fc5a50
447cf0ba089a8cacf6c14029c0bb255
5bcd4f18069050060a0e94dd2bdf1c21
34455d8d06da965fe4622e589fa8548c
4ded26663800dd624e7b294dd5ba92fe
c24bd727114da0864e336469e812293
e5768bb3cf72c5771eea84510df9948
2c634c0871757b7fe26d12fca080904e
7107bcl18ddefa81cb14323b703cc82
49a2e198b94de669f3631ce9848b842a
09afe7e8aa8eef37f1ce8a7d6c3a07af
9b9c2f85ee3d81e687cef677bd476b06
94610bfe79884b88d225fdaabb79ed37
754ed3655246a81c933b99d9abb75ce0
26cd6c368de07b45af60af21860152cf
5da03ebf6aecb43783ed214d2b366310
935e030937d3677923e25788c42c59d7
efa296a6c383b1e661f52139516a2f9
b6227690244cdc465541c05a53430ac4
cc5e6ac139359a21fcc3bdc6e2ddc143
84083d41fb1044f968bb3a018eab992d
07a5d3ce1dccl14e469ef01c012d4a12
85ecb492d2e756e2f0fc6db5bee713cc
2c3c299deb7b39c5a2406cae582cb224
```

Current Technical Challenges

- Measuring entropy of dataset
 - Exploring algorithms to quantify randomness of our dataset
 - Communicating with Mathematics department to ensure our approach is valid
- Implementing MD5 algorithm
- Creating interactive features of website for key attributes
- Fixing Webhooks bug
- Finding public domain, high resolution background images
- Overcoming the html & css learning curve
- Displaying data used for our encryption so it is visually engaging to the user

Milestone 1 Task Matrix

Task	Adonay	Jared	Josh	Luke
1. Update demos	25%	25%	25%	25%
2. Update Documentation	Read & Review	Read & Review	100%	Read & Review
3. Add Content to the Web App (About, Generate Key, Learn more)	Read & Review	Read & Review	75%	25%
4. Full functionality to Generate Key button (Strike info, md5)	50%	Offer help / troubleshoot	Offer help / troubleshoot	50%
5. Fix Webhook bug	Offer help / troubleshoot	Offer help / troubleshoot	100%	Offer help / troubleshoot
6. Create website domain name	Offer help / troubleshoot	Offer help / troubleshoot	Offer help / troubleshoot	100%
7. Create LinkedIn Profiles and link to Web App Team page	25%	25%	25%	25%

Milestone 1 Task Matrix

8. Automation that stores all generated numbers in database	50%	50%	Offer help / troubleshoot	Offer help / troubleshoot
9. Create documentation explaining the generation of key	Offer help / troubleshoot	50%	50%	Offer help / troubleshoot
10. Generate key from database, insert key into database, MD5 hash, display MD5 hash on website	50%	Offer help / troubleshoot	Offer help / troubleshoot	50%

Questions