CS Capstone Progress Report

December 4, 2018

Auction Hunter

Prepared for

Ryan Kalb

Prepared by

Group 4
Auction Hunter
Alexander Hull
Alexander Jacobson
Yufei Zeng

Abstract

Progress report which contains our teams successes, challenges, and work still needs to be completed. This encompasses the work completed during Fall 2018 term.

Contents

1 Introduction

Our Auction Hunter project is meant to make searching for salvaged car auctions much easier for the user. Auction Hunter aims to help users get a better deal and know what they are paying for. It is important to first establish a foundation of technologies, design, and implementation schedule before actual implementation should be completed. It is important that we minimize the amount of time spent working on a component that will eventually not work out, or using a technology that we later find won't interact with another piece. Because of this, only cursory implementation has been completed.

2 Project Status

The Auction Hunter team has finished our initial planning and is moving on to designing our architecture for our implementation. We have worked with our customer to determine exactly what features we want and which are the highest priority. The Auction Hunter team has spent much of the term researching and writing technical documents in preparation for starting the coding phase. We have also starting investigating how we want to organize and implement parts of the project such as the web crawler and database storage. Below is our initial flow diagram which lays out how we intend to build Auction Hunter in a piecemeal way. Each team members will be able to work on independent portions so as not to block each other from working.

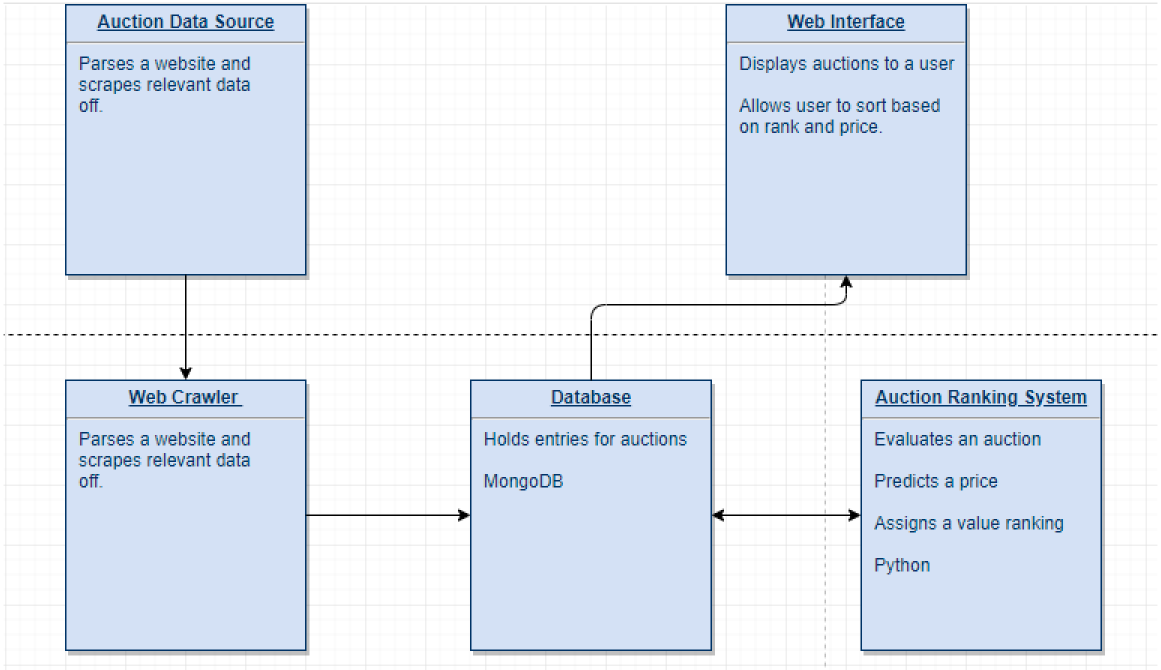

Auction Data Source
Parses a website and scrapes relevant data off.
Web Interface
Displays auctions to a user
Allows user to sort based on rank and price.
Web Crawler
Parses a website and scrapes relevant data off.
Database
Holds entries for auctions
MongoDB
Auction Ranking System
Evaluates an auction
Predicts a price
Assigns a value ranking
Python

Fig. 1. Flow Design

3 Obstacles

The biggest obstacle that we have faced thus far is communication. This includes communication with out client, instructors, and TA. We were able to improve over the course of the term, but we should have been more proactive in the first few weeks. Our group never got far off track, so the impact was minimal. However, there were a few points that could have been made easier if we had better communication.

Another obstacle was finding a way to arrive at a group decision on the choices of technology. Because the technology review was an individual assignment, we were unable to fully discuss the pros and cons of each choice with our team. In the end, our group is happy with the choices that were made individually.

4 Interesting Code

4.1 MongoDB database

Below is an example of how to create a new entry in Python using a MongoDB database [1]. We can create a uniform auction entry template, then fill in relevant data that we pull from the website. Ideally we will modify this template to match 1:1 with all the important data that the auction websites provide. It may be the case that we don't end up using every piece of data, but it is important to not limit ourselves later down the pipeline.

```
db.inventory.insertOne(
   { item: "Tesla Model S", price: 12500, tags: ["bumper damage"] }
)
```

Fig. 2. MongoDB 'insert' example

Below is an example python call that can be made using MongoDB to find a particular subset of our database. In this example, we would find all entries with a price greater than $10,000. This will be a crucial component when we go to implement our website UI and value calculation components.

```
db.collection.find( { price: { gt: 10000 } } )
```

Fig. 3. MongoDB 'find' example

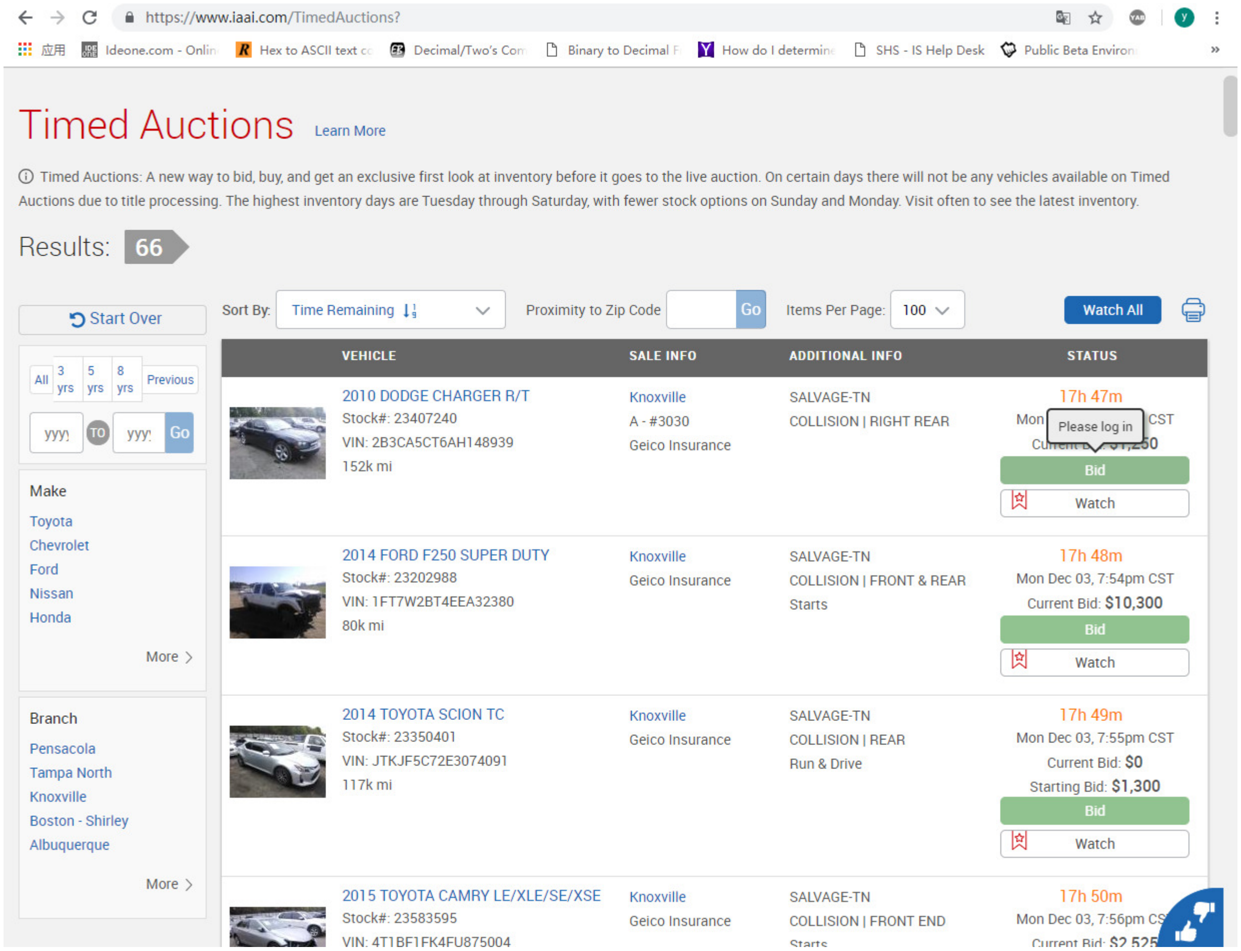
4.2 Web Scraper

When customers are trying to buy a salvage car and they will want to see the prices at different platforms at a single place. In such situations, Web crawler will full play its function on building an aggregator platform.

IAAAI.com is a website that sells salvage cars. [2] It will be our test site, and we are going to scrape Timed Auctions for salvage cars. The first step is to generate a basic spider using python:

```
scrapy genspider timedauctions https://www.iaai.com/TimedAuctions
```

That is what the IAAAI Timed Auctions page looks like:



Timed Auctions

ⓘ Timed Auctions: A new way to bid, buy, and get an exclusive first look at inventory before it goes to the live auction. On certain days there will not be any vehicles available on Timed Auctions due to title processing. The highest inventory days are Tuesday through Saturday, with fewer stock options on Sunday and Monday. Visit often to see the latest inventory.

Results: 66

Sort By: Time Remaining | Proximity to Zip Code | Items Per Page: 100

Make: Toyota, Chevrolet, Ford, Nissan, Honda

Branch: Pensacola, Tampa North, Knoxville, Boston - Shirley, Albuquerque

	VEHICLE	SALE INFO	ADDITIONAL INFO	STATUS
	2010 DODGE CHARGER R/T Stock#: 23407240 VIN: 2B3CA5CT6AH148939 152k mi	Knoxville A - #3030 Geico Insurance	SALVAGE-TN COLLISION | RIGHT REAR	17h 47m Mon CST Cu 50
	2014 FORD F250 SUPER DUTY Stock#: 23202988 VIN: 1FT7W2BT4EEA32380 80k mi	Knoxville Geico Insurance	SALVAGE-TN COLLISION | FRONT & REAR Starts	17h 48m Mon Dec 03, 7:54pm CST Current Bid: $10,300
	2014 TOYOTA SCION TC Stock#: 23350401 VIN: JTKJF5C72E3074091 117k mi	Knoxville Geico Insurance	SALVAGE-TN COLLISION | REAR Run & Drive	17h 49m Mon Dec 03, 7:55pm CST Current Bid: $0 Starting Bid: $1,300
	2015 TOYOTA CAMRY LE/XLE/SE/XSE Stock#: 23583595 VIN: 4T1BF1FK4FU875004	Knoxville Geico Insurance	SALVAGE-TN COLLISION | FRONT END Starts	17h 50m Mon Dec 03, 7:56pm CS Current Bid: $2,525

Fig. 4. Timed Auctions Page

From this page, the following data about a salvage car needs to be extracted:

- Salvage car photo
- Salvage car current bid
- Salvage car VIN
- Salvage car time remaining
- Salvage car price

Extracting the URLs include salvage car photo:

After searching through the web page source code by using the developer tools of Chrome. We found that the car photos are stored under vis.iaai.com with imageKeys.

The attribute ”imageKeys” can be used to extract image URLs.

```
response.css("img::attr(imageKeys)").extract()
```

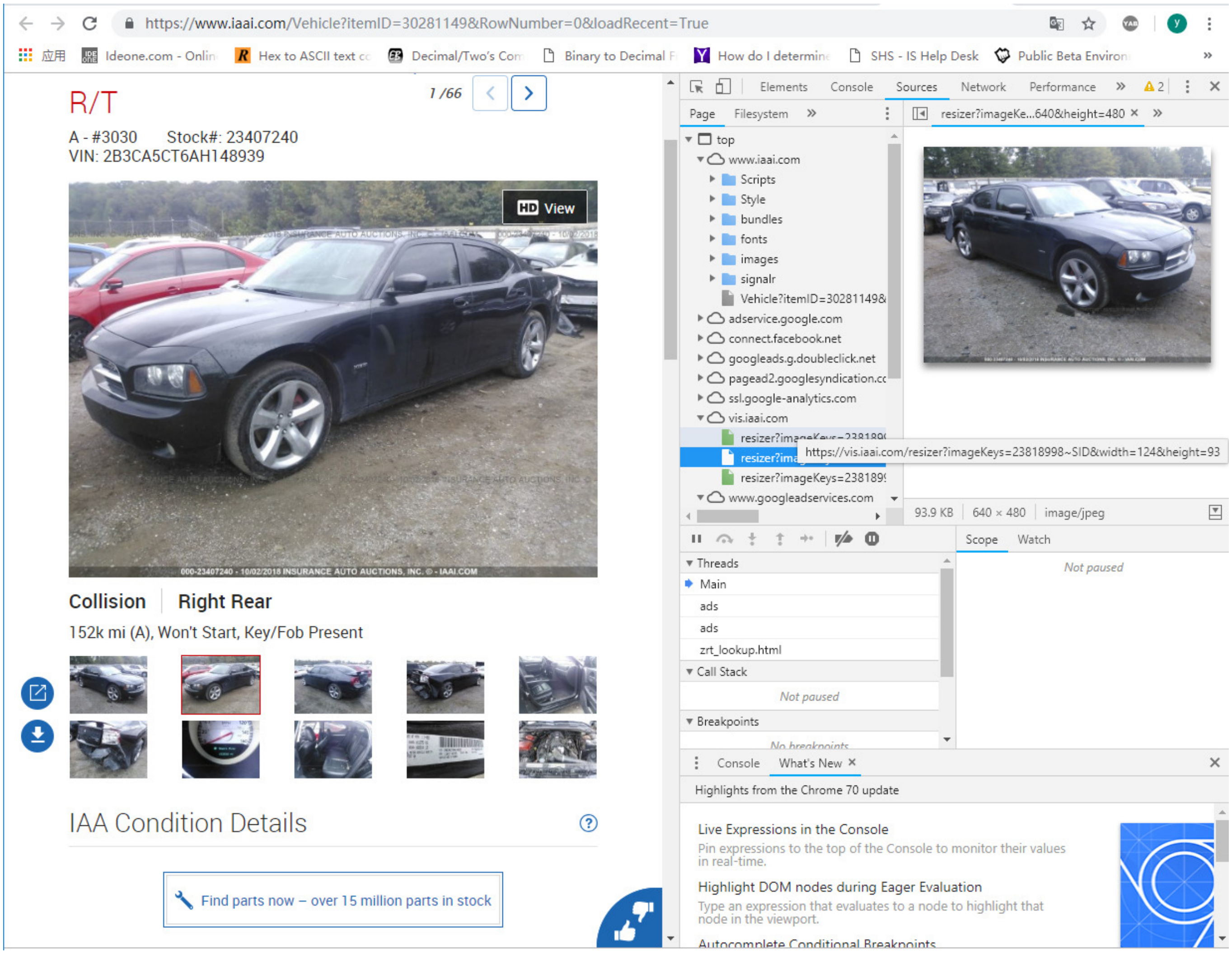

https://www.iaai.com/Vehicle?itemID=30281149&RowNumber=0&loadRecent=True
应用
Ideone.com - Onlin
Hex to ASCII text c
Decimal/Two's Con
Binary to Decimal F
How do I determine
SHS - IS Help Desk
Public Beta Environ
R/T
1 /66
A - #3030 Stock#: 23407240
VIN: 2B3CA5CT6AH148939
HD View
Collision
Right Rear
152k mi (A), Won't Start, Key/Fob Present
IAA Condition Details
Find parts now – over 15 million parts in stock
Elements
Console
Sources
Network
Performance
Page
Filesystem
resizer?imageKe...640&height=480
top
www.iaai.com
Scripts
Style
bundles
fonts
images
signalr
Vehicle?itemID=30281149&
adservice.google.com
connect.facebook.net
googleads.g.doubleclick.net
pagead2.googlesyndication.cc
ssl.google-analytics.com
vis.iaai.com
resizer?imageKeys=238189
resizer?ima
https://vis.iaai.com/resizer?imageKeys=23818998~SID&width=124&height=93
resizer?imageKeys=238189
www.googleadservices.com
93.9 KB
640 × 480
image/jpeg
Scope
Watch
Not paused
Threads
Main
ads
ads
zrt_lookup.html
Call Stack
Not paused
Breakpoints
No breakpoints
Console
What's New
Highlights from the Chrome 70 update
Live Expressions in the Console
Pin expressions to the top of the Console to monitor their values in real-time.
Highlight DOM nodes during Eager Evaluation
Type an expression that evaluates to a node to highlight that node in the viewport.
Autocomplete Conditional Breakpoints

Fig. 5. Salvage Car Photo

Extracting the URLs include salvage car current bid:

We can use a similar method to find VIN's location which is an attribute of the <tbody>tag.

```
response.css("tbody::attr(vin)").extract()
```

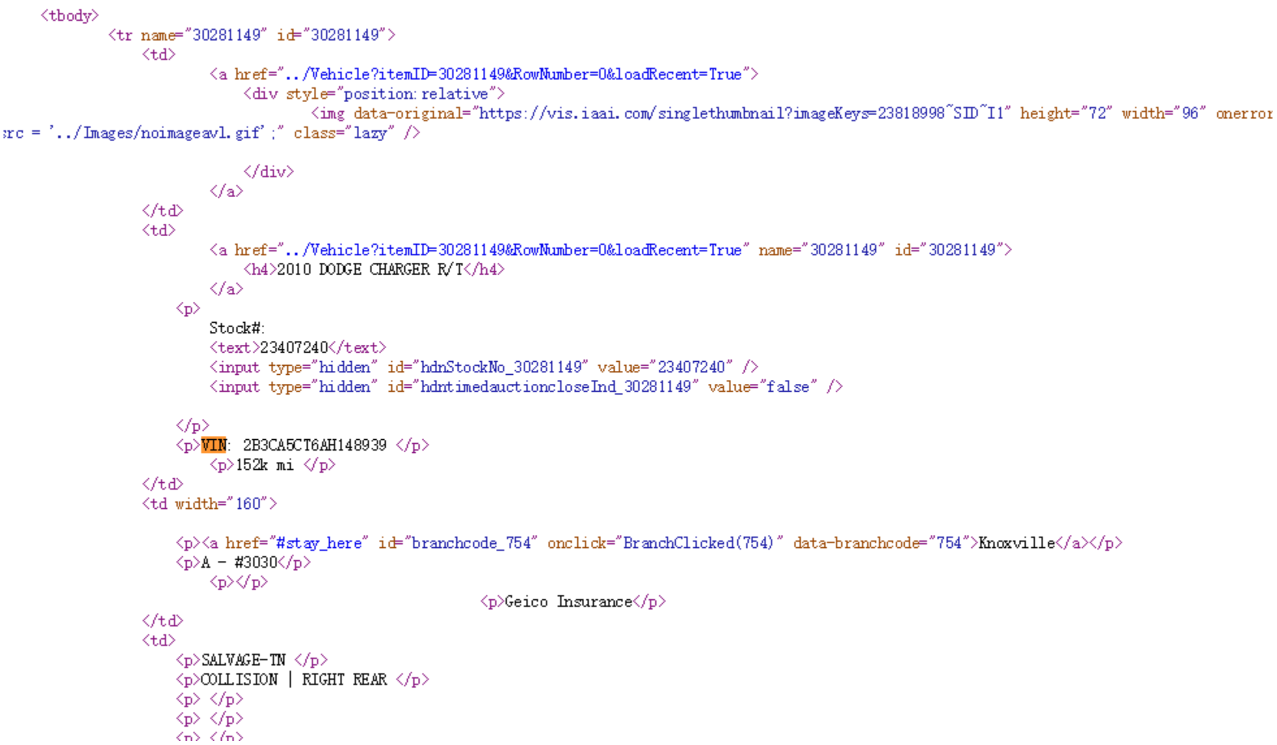
Extracting the URLs include salvage car remaining time and price:

Similarly, we are able to extract the remaining time and price of salvage car.

Time to download the extracted photos of salvage car:

As mentioned in Design Document, Scrapy provides the images pipelines: once we got the data from website, we are able to pass them through different pipelines. By taking advantage of this function, the image pipeline allows us to download extracted photos of salvage car. In addition, we are able to convert the format of images and generate thumbnail.

```
#enable the images pipeline.
SalvageCarPhotos_PIPELINES='scrapy.pipelines.images.ImagesPipeline': 1
#set the local download address.
```



```
     <tbody>
             <tr name="30281149" id="30281149">
                 <td>
                         <a href="../Vehicle?itemID=30281149&RowNumber=0&loadRecent=True">
                             <div style="position: relative">
                                     <img data-original="https://vis.iaai.com/singlethumbnail?imageKeys=23818998~SID~I1" height="72" width="96" onerror
rc = '../Images/noimageavl.gif';" class="lazy" />

                             </div>
                         </a>
                 </td>
                 <td>
                         <a href="../Vehicle?itemID=30281149&RowNumber=0&loadRecent=True" name="30281149" id="30281149">
                             <h4>2010 DODGE CHARGER R/T</h4>
                         </a>
                     <p>
                         Stock#:
                         <text>23407240</text>
                         <input type="hidden" id="hdnStockNo_30281149" value="23407240" />
                         <input type="hidden" id="hdntimedauctioncloseInd_30281149" value="false" />

                     </p>
                     <p>VIN: 2B3CA5CT6AH148939 </p>
                         <p>152k mi </p>
                 </td>
                 <td width="160">

                     <p><a href="#stay_here" id="branchcode_754" onclick="BranchClicked(754)" data-branchcode="754">Knoxville</a></p>
                     <p>A - #3030</p>
                         <p></p>
                                                     <p>Geico Insurance</p>
                 </td>
                 <td>
                     <p>SALVAGE-TN </p>
                     <p>COLLISION | RIGHT REAR </p>
                     <p> </p>
                     <p> </p>
                     <p> </p>
```

Fig. 6. Salvage Vin Photo

```
Photo_Store = 'Users/Desktop/SalvageCarPhotos/'
#Generate two kinds of thumbnail for each salvage car photo, one small, one large.
GenerateThumbnail = {'small': (20, 20), 'large': (100, 100),}
```

5 Retrospective

Week	Positives (What we completed)	Deltas (What we need to complete)	Actions (How we are going to complete them)
1	Fictional Biography (individual assignment for self-introduction.), Resume (individual assignment for applying jobs or graduate school.)	We will individually need to complete the resume peer review.	We all have partially finished resumes already, so it is a matter of updating.
2	Select a project (individual assignment.)	We will individually need to decide on a project, and complete the project selection quiz.	We will be able to contact a number of clients to get in touch prior to the selection to see if the project is the right fit.

3	Problem Statement (individual assignment, including the details from client meeting and project proposal.)	We plan to complete the group problem statement tonight. We also want to look into other forms of communication so my group can collaborate on a project at the same time while working remotely.	For the time being we will use Discord for real time communication, and Slack for all other communication.
4	Group Problem Statement(group assignment, combining each group member's problem statement into one document.)	We will need to start looking at the requirements document and work to get in better contact with our client.	We will make sure that our client is in our Slack channel so he will stay closer in the loop and we can have a place to ask questions. We will discuss some project details with our client to make sure we add them to the requirement document.
5	Our team has drafted up the majority of our requirements document. We are now waiting to hear back from our client to get some feedback and/or approval.	We plan to complete the requirements document this weekend, after we hear back from our client.	We have contacted our client over email, although we had the wrong address. We were able to resolve this over slack.
6	Our group recently finished the requirements document. We collaborated well for this last assignment and things went well. Our group also got together and assigned portions of the project for the tech review.	We will start looking at available technologies and begin testing.	For each task that needs to be completed, we can look into sample code or technologies that could fulfill our requirements.
7	Our group is working on applying all feedback from the tech review peer review.	We are working on completing our final tech reviews.	The combination of the two reviews will be used to edit our technology review drafts to work towards a final. The main points of concern are being less vague.
8	We have finished the technology review and agreed on the technologies to use.	We will need to learn more about the technologies that we chose, how to effectively use them, and putting the pieces together.	Looking at example code is a good place to start. Since our basic needs are fairly simple, we can research starter code to get an idea of what needs to be done.

9	We finished some initial technology investigation to see what the implementation would involve. We generated some code snippets.	We plan to begin working on the design document soon.	Since we are all celebrating the holiday weekend, we likely won't putting a lot of work into it until the Sunday. Thankfully there are large individual components that we can all think about before starting.
10	We finished the design document, and were able to pull information from our technology reviews.	Looking forward, the progress report and client verification still needs to be completed.	We agreed that we will all work on the written progress report, then copy a lot of the info to the presentation. To make recording easier, we will have two members create the presentation and one member speak over it.

REFERENCES

[1] MongoDB, "What is mongodb," https://www.mongodb.com/what-is-mongodb, 2018, accessed: 2018-11-9.

[2] IAAI, "Faq," https://www.iaai.com/Support/SupportFaq.aspx, 2018, accessed: 2018-11-9.