

## SE G15: Individual Contribution

- Following tasks are done by the team together.

- Requirement Elicitation
- Choosing suitable Process Model
- Making Use Case Models
- Requirement Analysis
- Activity Diagram

### Build 1: User Authentication **20 March - 11 April**

#### Iteration 1: 20 - 27 March

- Implemented User Registration through Email ID, Password, Confirm-Password
- Implemented Login through Email ID, Password
- Generating 64 char long Unique API\_KEY on successful registration
  - Used UUIDField Model available in Django to get **universally unique identifiers**.

#### Iteration 2: 1 - 11 April

- Implemented Password Reset Functionality
  - We are sending an email containing a unique URL to the registered Email ID, from that URL user can reset the password.
- Implemented Password Change
  - Logged In users can change password from Profile Page by providing a current and new password.

Contributor	Issue	PR
Deven	<a href="#">#16</a>	Initial Commit
Paresh	<a href="#">#5</a>	<a href="#">#17</a> , <a href="#">#18</a>
Tanu	<a href="#">#5</a>	<a href="#">#28</a>

### Build 2: Scraper **27 March - 3 May**

#### Iteration 1: 27 March - 16 April (Suspended In-between)

- Build Specific Scrappers to scrape particular sites.
  - We analyzed various web pages' HTML code and it turns out that almost every site has a different structure for presenting the page content. So we have decided to build targeted scrappers for popular sites with a 100% current data guarantee, and one generic scraper that will try to scrape any page that doesn't have a targeted scraper. So generic scraper might have a lesser guarantee for current data retrieval.

Iteration 2: 16 April - 20 April

- Implemented Scraper using metadata parser

Iteration 3: 23 April - 3 May

- Improve performance of Scraper

Contributor	Issue	PR
Paresh	<a href="#">#22</a>	<a href="#">#26</a>
Isha	<a href="#">#22</a>	<a href="#">#34</a> , <a href="#">#35</a>
Manan	<a href="#">#1</a>	<a href="#">#43</a> , <a href="#">#57</a>

## Build 3: API End-Point 10 April - 1 May

Iteration 1: 10 April - 20 April

- Build Model Schema for API response
- Build serializer for the above Model
- Created API End-Point
- Returning dummy data as of now
  - Scraper was not fully functional

Iteration 2: 20 April - 1 May

- Connected Scraper
- Started Caching the responses
- Utilizing Cache data and calling scraper only in required scenarios

Contributors	Issue	PR
Shivam & Keyur	<a href="#">#3</a>	<a href="#">#42</a> , <a href="#">#45</a>

## Build 4: Frontend 5 April - 20 April

Iteration 1: 5 - 12 April

- Build Design Mockup

Iteration 2: 13 - 20 April

- Implemented Frontend Pages according to the mockup
  - Home Page
  - Login Page
  - Registration Page
  - Profile Page
  - Password Reset/Change Page
  - Documentation Page

Contributor	Issue	PR
Harshil	<a href="#">#2</a>	<a href="#">#32</a>
Manan	<a href="#">#2</a> , <a href="#">#63</a>	<a href="#">#41</a> , <a href="#">#64</a>
Milan	<a href="#">#2</a>	<a href="#">#33</a>
Keyur	<a href="#">#2</a>	<a href="#">#21</a> , <a href="#">#36</a> , <a href="#">#37</a>
Shruti	<a href="#">#2</a>	<a href="#">#38</a>

## Build 5: Cache & Cron Jobs **15 April - 5 May**

- Implemented Daily Limit Reset Crawler
- Implemented Cache Maintenance Crawler
- Implemented Cache

Contributor	Issue	PR
Tanu & Shruti	<a href="#">#6</a>	<a href="#">#44</a>
Harshil	<a href="#">#47</a>	<a href="#">#46</a>

## Testing

- GUI testing  
Contributors: Deven, Paresh
- Unit testing  
Contributor: Shivam
- Stress testing  
Contributor: Shivam
- Black box testing  
Contributor: Harshil
- Acceptance test  
Contributor: Isha

## Other Tasks

- Deployment  
Contributors: Milan, Harshil
- API Documentation  
Contributor: Isha
- Design Documents  
Contributors: Keyur, Milan
- Final SRS, Minutes of Meeting  
Contributor: Shruti
- Demo Video  
Contributor: Tanu

## Member-wise Contribution

[Shivam](#) : API Endpoint, Unit testing, Stress testing

[Milan](#) : Frontend (Base, Home), Design Document, Deployment

[Isha](#) : Web scraper (Flipkart), Acceptance testing, API documentation

[Keyur](#) : Frontend (Login, Register), API Serializer, Design Document

[Tanu](#) : Password Change/reset, Daily Limit reset cronjob, Demo video

[Manan](#) : Frontend (Password change\reset, API doc. page), Web Scraper (Iteration 2)

[Harshil](#) : UI design, Cache Implementation, Black box testing, Deployment

[Paresh](#) : Password Change/reset, Web Scraper (Wikipedia), GUI testing

[Deven](#) : User Authentication, GUI testing, Module Integration

[Shruti](#) : Frontend (Account), Cache Maintenance cronjob, SRS documentation, Minutes of meeting

Github repo: <https://github.com/WebMetadataRetrieval>