

# **E-Commerce Website of Books**

*Project report submitted  
in partial fulfillment of the requirement for the degree of*

**Bachelor of Technology**

By

**Aayush Gadia**

(Reg No. 100436 of 2016-17)

**Gautam Anand**

(Reg No. 100437 of 2016-17)



Dept. of Engineering & Technological Studies  
University of Kalyani, Nadia, West Bengal

July 2020

## **CERTIFICATE**

It is certified that the work contained in the project report titled “Ecommerce Website of Books”, by “Aayush Gadia & Gautam Anand”, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree

**Signature of Supervisor**  
**Dr. Sanjoy Das**  
**Senior Scientific Officer**  
**Dept. of Engineering & Technological Studies**  
**University of Kalyani**  
July, 2020

## Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

*Aayush Gadia*

(Signature)

**AAYUSH GADIA**

(Name of the student)

**100436 of 2016 - 17**

(Reg No.)

Date: **11.07.2020**

*Gautam Anand*

(Signature)

**GAUTAM ANAND**

(Name of the student)

**100437 of 2016 - 17**

(Reg No.)

Date: **11.07.2020**

# INDEX

TOPIC	PAGE NO.
Acknowledgment	5
Project Objective	6 – 7
Required Specifications	8
Frontend Design	9 – 16
Database Design	17 – 24
Backend API's	25 – 44
Chatbot	45 – 58
Server Configuration	59 - 60

## Acknowledgment

I take this opportunity to express my profound gratitude and deep regards to my supervisor Dr. Sanjoy Das for providing their invaluable guidance, comments and suggestions throughout the course of the project.

The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark. I am obliged to my team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the entire period.

Aayush Godia

Gautam Anand

## Project Objective

The project is on **E Commerce website** of only **books**. It would actually be a portal where a **user can buy new books** as well as **old books** and they can also **donate books**. The website would be having all the features that are found in big e-commerce websites like Amazon, Flipkart etc.

The **motive** behind the project was, that, we **observe that at homes after the session year of school ends, parents sell the books of their children to the scrap collector**. So, what if there was a facility where people can **donate their books so that those books can be used for someone who actually can't afford to buy new one**.

In case of **Old Books**-

- Our portal provides the **facility of donating books**, so what parents or anyone who wants to donate their old books can do is that, they can **simply place an order of donating books** to us by simply filling a form mentioning all the necessary details.
- Once, the **order has been placed**, one of our delivery partners would **pick up the books** from the address mentioned making sure that the **books donated meets the certain quality standards**, if it doesn't meet it then it won't be picking up the order.
- Then, the **donated books details are entered into the database** of our system by our team.
- That's it, and now the **donated books are live on the website and anyone can buy it for free by just paying the delivery charge**.
- If there are **many books of a particular ISBN with varying books conditions**, then all the conditions are shown on the product page of that ISBN's book. Buyer can simply choose the condition they want, check the price and can simply place an order.

In case of **New Books**-

- Our portal also **provides the buyers an option to buy new books.**
- It might not be that for each new book on the portal, there exists a donated (used) book with varying conditions like **readable, very good, almost new & brand new.**

**Similarly, for each donated (used) book it might not be that there exists a brand new book on the portal.**

- The **new books are not free at all, and along with the book price, buyer has to pay the delivery charge according to the weight of the books & the delivery pincode.**
- If there are many books of a particular ISBN with varying books conditions, then all the conditions are shown on the product page of that ISBN's book. Buyer can simply choose the condition they want, check the price and can simply place an order.

Thus our **motive is to provide free books to those who have the desire to read but lack the resources to buy.** With advancement in Internet & Technology thus anyone **can go to our website and place a request for any book displayed on the website.**

Also, when books are printed and published they release an average of **2.71 kgs of CO<sub>2</sub> equivalent**, therefore, if we **simply re-use the books then we are saving the CO<sub>2</sub> released & we are also saving a lot of trees indirectly.** Therefore, our motive is environment friendly and is contributing somehow in healing our planet.

## **Required Specifications**

Following are the required specifications-

1. HTML5
2. CSS3
3. JavaScript 1.8.5
4. ReactJS 16
5. Django 2.1.5
6. MySQL 5.6
7. Python 3.x
8. TypeSense Server 0.14.0
9. Rasa X 0.27.2
10. RazorPay

# FrontEnd Design

The screenshot shows a website for buying books. At the top, there's a navigation bar with links for 'Categories', 'Search Books, Author, Publisher, Title, ISBN #', 'Donate Book', 'Log In/Sign Up', 'Cart', and 'Offers'. Below the search bar is a dropdown menu for 'Competitive Exams' with options like 'CAT', 'Engineering & Medical', 'Govt Jobs', 'GK', 'International Exams', and 'School Level', along with a link to 'View All Competitive Exams Books'. The main content area displays five book covers in a grid:

- Hindi Vyakaran (₹99- FREE\*)
- A TEXTBOOK OF PHYSICS 2ND YEAR PRE-UNIVE... (₹270- FREE\*)
- IBS INSPIRATIONAL BOOK OF SCIENCE (₹50- FREE\*)
- MTG 30 YEARS CHAPTERWISE SOLUTIONS IIT-JEE MATHMATICS (₹200- FREE\*)
- ARIHANT OBJECTIVE PHYSICS IIT SCREENING (₹170- FREE\*)

Below this, another section for 'Fiction & Non Fiction' is shown with five more book covers:

- KRIYA YOGA (₹445- FREE\*)
- Travelling Light : Walking The Path Of L... (₹195- FREE\*)
- Know Your Child : Art Of Raising Children (₹129- FREE\*)
- Himachal (₹295- FREE\*)
- MERA BACHPAN (₹120- FREE\*)

On the right side of the page, there's a blue circular icon with a white '2' inside, likely indicating a notification or message count.

Fig 1 – HomePage

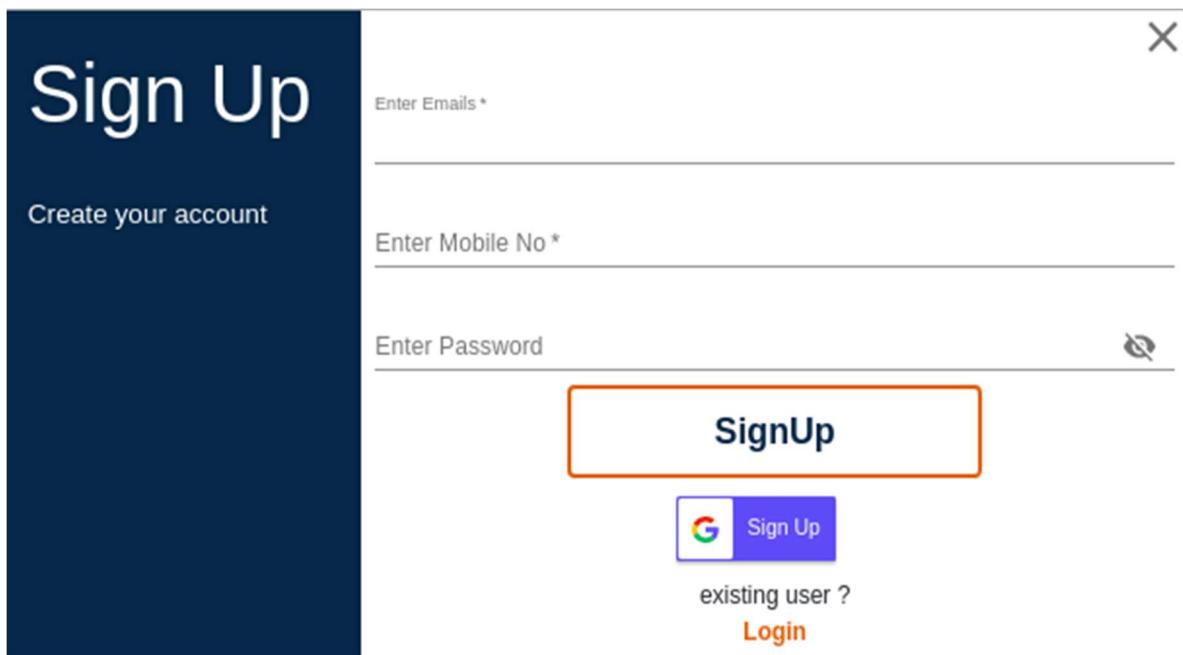
**Figure 1**, is the **Home Page of the website**. This page shows the books in different categories like **Competitive Exams, Fiction & Non Fiction, School & Children Books, University Books**. Also, this page has options to do **Login or Sign Up to the website**. There is a feature called **Donate Books** which anyone can use to donate their books. There is a **search bar** on the top of the page where one can **search a book by its name or author name or publishing house or by the ISBN number**.

## Competitive Exams

<b>Engineering &amp; Medical</b>	<b>School Level</b>	<b>Non Fiction</b>	<b>Children books</b>	<b>NCERT</b>	<b>Medical</b>
AIEEE	Navodaya Vidyalaya	Biographies	Action & Adventure	Class 6	Ayurveda
Engineering(Post Graduate)	NTSE	Coffee Table	Activity Books	Class 7	Dental
IIT	Olympiads	Computer & Internet	Children's literature	Class 8	MBBS
Medical	Others	Cooking	Comics & Graphic Novels	Class 9	Others
Others	Sainiks	Entertainment	Disney Store	Class 10	PG
State Level		House & Home	Fun & humor	Class 11	Pharmacy
<b>Government Jobs</b>		Pets	History & Mythology	Class 12	<b>Others</b>
Banking	<b>Comics &amp; Graphic novels</b>	Photography	Knowledge & Learning	<b>CAT(MBA)</b>	BBA
Others	<b>Drama</b>	Sports	Others		B Com
Railway	Horror & Thriller	Travel	Picture Book		Law
SSC	Literary		<b>Classes</b>		MBA
Teaching Related Exams	Mystery	Holy Books	KG		M Com
UPSC	Romance	Others	Class 1		Others
International Exams	Science Fiction	philosophy	Class 2		Probability & Statistics
GMAT	Short Stories	Religions	Class 3		PhD
GRE	Teens	Spirituality	Class 4		<b>Science &amp; Arts</b>
Others	<b>Fiction &amp; Non Fiction</b>	<b>Motivation &amp; Self-help</b>	Class 5		B.A.
SAT	others	<b>School &amp; Children Books</b>	Class 6		BCA
General knowledge & Learning	Astrology	<b>Dictionary Level</b>	Class 7		BSc
	Business & Management	English To English	Class 8		M.A.
	Health	English To Hindi	Class 9		MCA
	History & Politics	Foreign Language	Class 10		MSC
	Others	Hindi To English	Class 11		
	Sports & Games	Subject Wise	Class 12		

Fig 2 – Books Categories Map

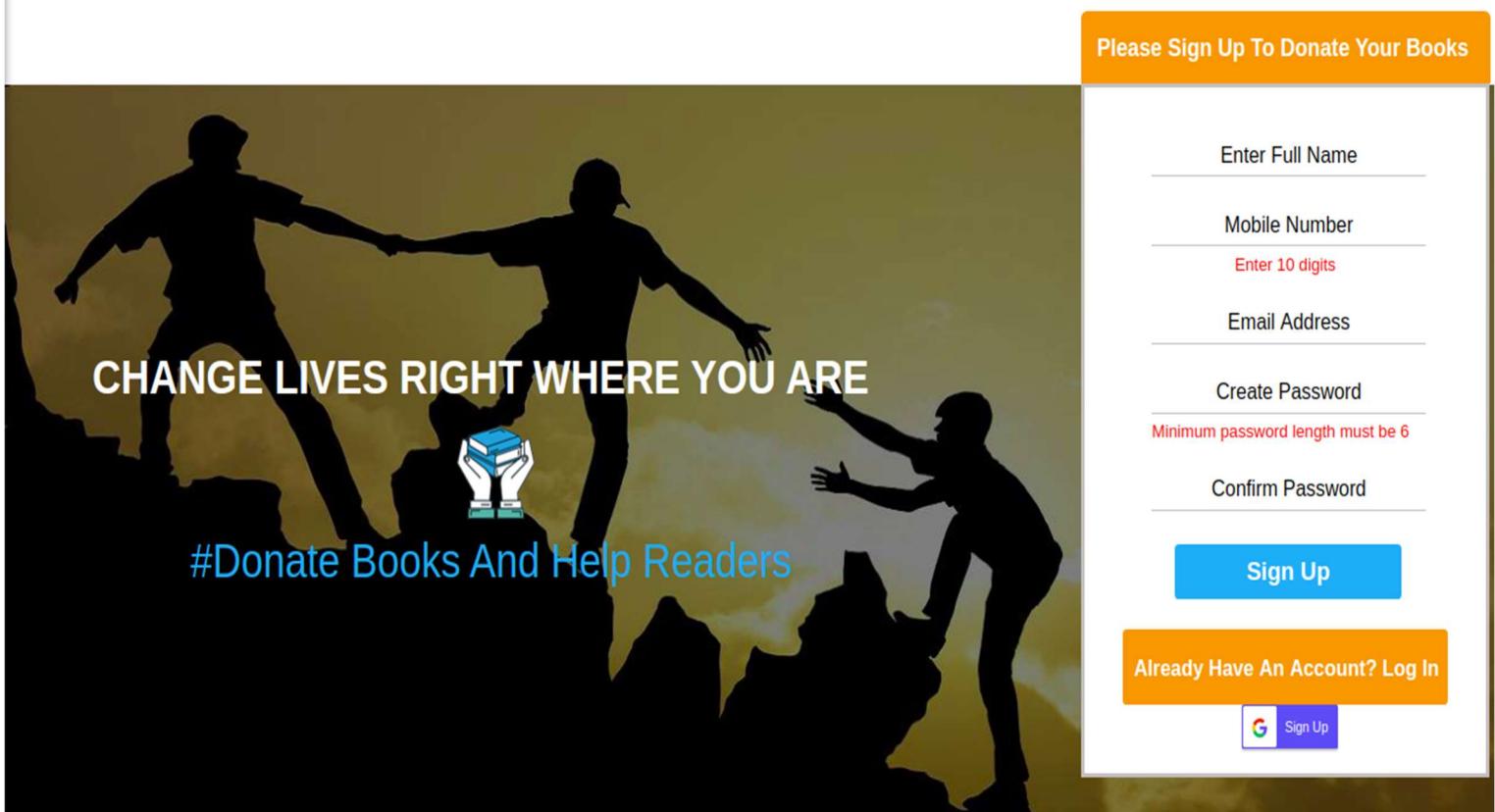
**Figure 2,** shows the different categories & their sub-categories in the books section viz implemented in the website.



The image shows a 'Sign Up' page with a dark blue header and a white form area. The header has the text 'Sign Up' and 'Create your account'. The form includes fields for 'Enter Email \*', 'Enter Mobile No \*', and 'Enter Password'. Below the password field is a 'SignUp' button with an orange border. At the bottom left is a Google sign-in button with the text 'Sign Up'. To the right of the button are links for 'existing user ?' and 'Login'.

Fig 3 – Sign Up / Login Page

**Figure 3,** is the Login / Sign Up Page.



**Fig 4 – Donate Books Page**

This screenshot shows the same sign-up form after logging in. The user profile information is displayed on the left, and the donation details are on the right.

User Profile Information (Left):

- Full Name
- Mobile Number (Enter 10 Digits)
- Address 1
- Address 2
- Landmark
- Pincode
- City
- State
- No Of Books You Want To Donate
- No Of Cartons/Boxes (In Which Books To Be Filled)
- Approx. Weight Of Books (Min 15Kg)

Select Donated Book Categories (Right):

- NCERT
- Novel
- Engineering
- Medical
- School
- Children
- MBA
- Study package
- Entrance exam
- Others

From Where You Got To Know (Right):

- Facebook
- Instagram
- google search
- friends
- Quora
- others

Book Condition (Right):

- Almost new
- good & readabl
- slightly marked
- slight damage

**Fig 5 – Donate Books Page after Logging In**

**Figure 4** is the **Donate Books Page**, one can use this page to Donate Books. **Figure 5** is the **Donate Books Form** which opens up after the **User Logs-In or Signs-up or is already Logged-In**. This page could be used to fill the details of the Books which is being donated.

Filter 2

competitive-exams > engineering

<b>BOOK TYPE</b>					
<input checked="" type="checkbox"/> Used (Donated Book)					
<b>BINDING COVER</b>					
<input type="checkbox"/> hardback					
<input type="checkbox"/> Paperback					
<b>BOOK CONDITION</b>					
<input type="checkbox"/> readable					
<input type="checkbox"/> very good					
<input type="checkbox"/> almost new					
<input type="checkbox"/> brand new					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<input type="checkbox"/>					
<b>LATEST ARRIVALS</b>					
<input type="radio"/> Last 7 Days					
<input type="radio"/> Last 30 Days					
<input type="radio"/> Last 180 Days					
<input checked="" type="radio"/> Till Today					
<b>SUBJECT</b>					
<input type="checkbox"/> none					
<b>LANGUAGES</b>					
<input type="checkbox"/> English					
<input type="checkbox"/> bengali					
<input type="checkbox"/> hindi					
<input type="checkbox"/> foreign language					

13 Years Objective Solved Qu...  
₹205 - FREE

physics for jee and other en...  
₹380 - FREE

Formulae Chemistry:  
Formulae...  
₹65 - FREE

Physics IIT JEE CHAPTER WISE...  
₹210 - FREE

errorless physics volume 2 ...  
₹675 - FREE

algebra-1: course in mathema...  
₹200 - FREE

plane trigonometry part 1 cl...  
₹95 - FREE

course in mathematics for iit-jee  
₹650 - FREE

physics for you solved paper...  
₹20 - FREE

arhant objective physics ii...  
₹170 - FREE

**Fig 6 – Category Page**

**Object-Oriented Programming in Turbo C++ (English)**  
Author : Robert Lafore,

Language:English      Format:Paperback  
Weight:1.27 Kg

Share

**Used Book**

MRP : 279 (FREE)  
Pay Only for Shipping & Handling ₹143

Select Book Condition  
 Very Good (1 Books in stock)

Total Amount: ₹143  
Delivered in 4-7 days after dispatch

**ADD TO CART**

**Fig 7 – Product Page**

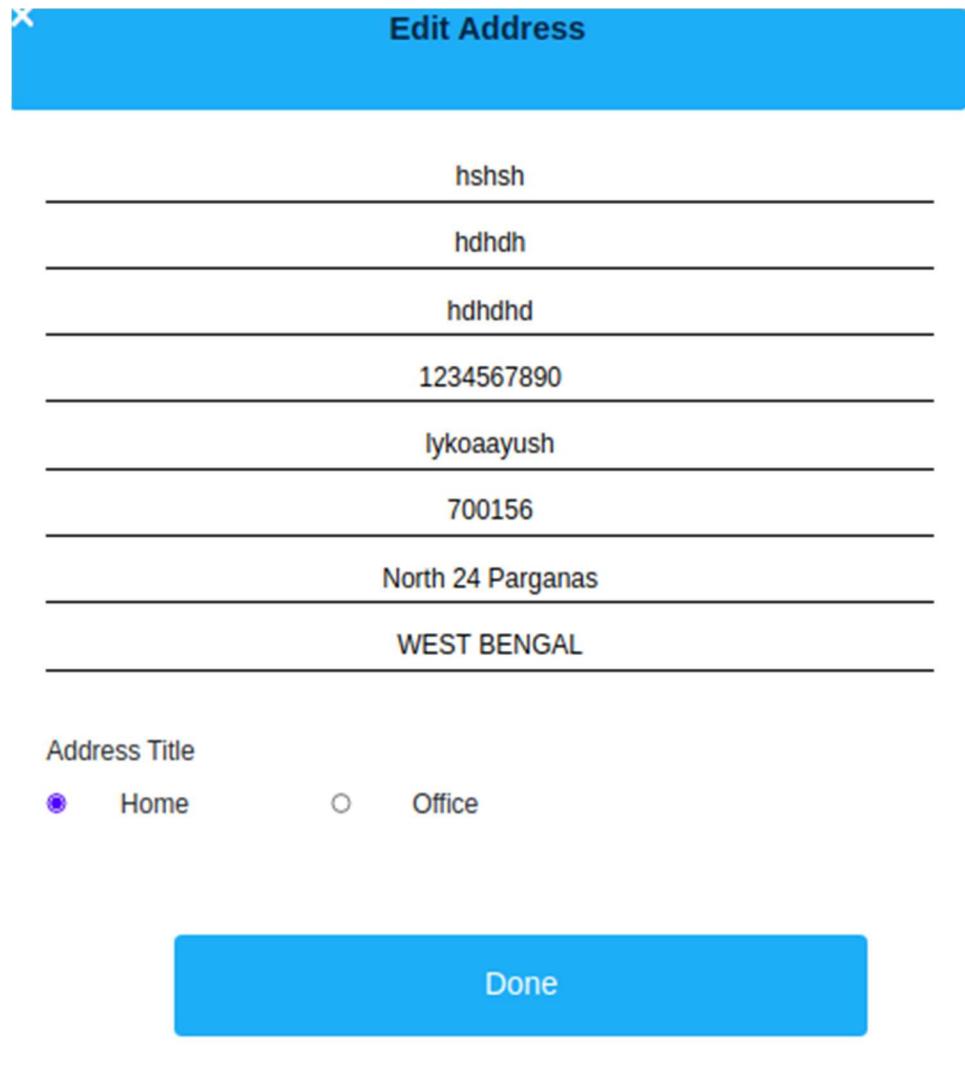
**Figure 6**, is how a **Categories Page** look like, let's say if someone clicks on Competitive Exams category or Engineering / Medical sub-category or AIEEE sub-sub-category then this is the layout of how the page would load up. **Figure 7**, is the **Product Page of a Book**.

The screenshot shows a mobile-style form titled "Add New Address". It consists of several input fields and a radio button group. At the top is a blue header bar with the title. Below it are seven horizontal input fields with labels: "Full Name", "Address 1", "Landmark", "Mobile Number", "Alternate Mobile Number Optional", "Pincode", and "State". Underneath these is a section labeled "Address Title" with two radio buttons: "Home" and "Office". At the bottom is a large blue "Done" button.

Field	Label
1	Full Name
2	Address 1
3	Landmark
4	Mobile Number
5	Alternate Mobile Number Optional
6	Pincode
7	State
8	Address Title
9	Home
10	Office
11	Done

**Fig 8 – User Address Add Page**

**Figure 8**, is used by the **user who has logged-in**, to add address and save them. The address added here could be **used later** by the user while placing an order. This would **prevent them from entering a details again & again**.



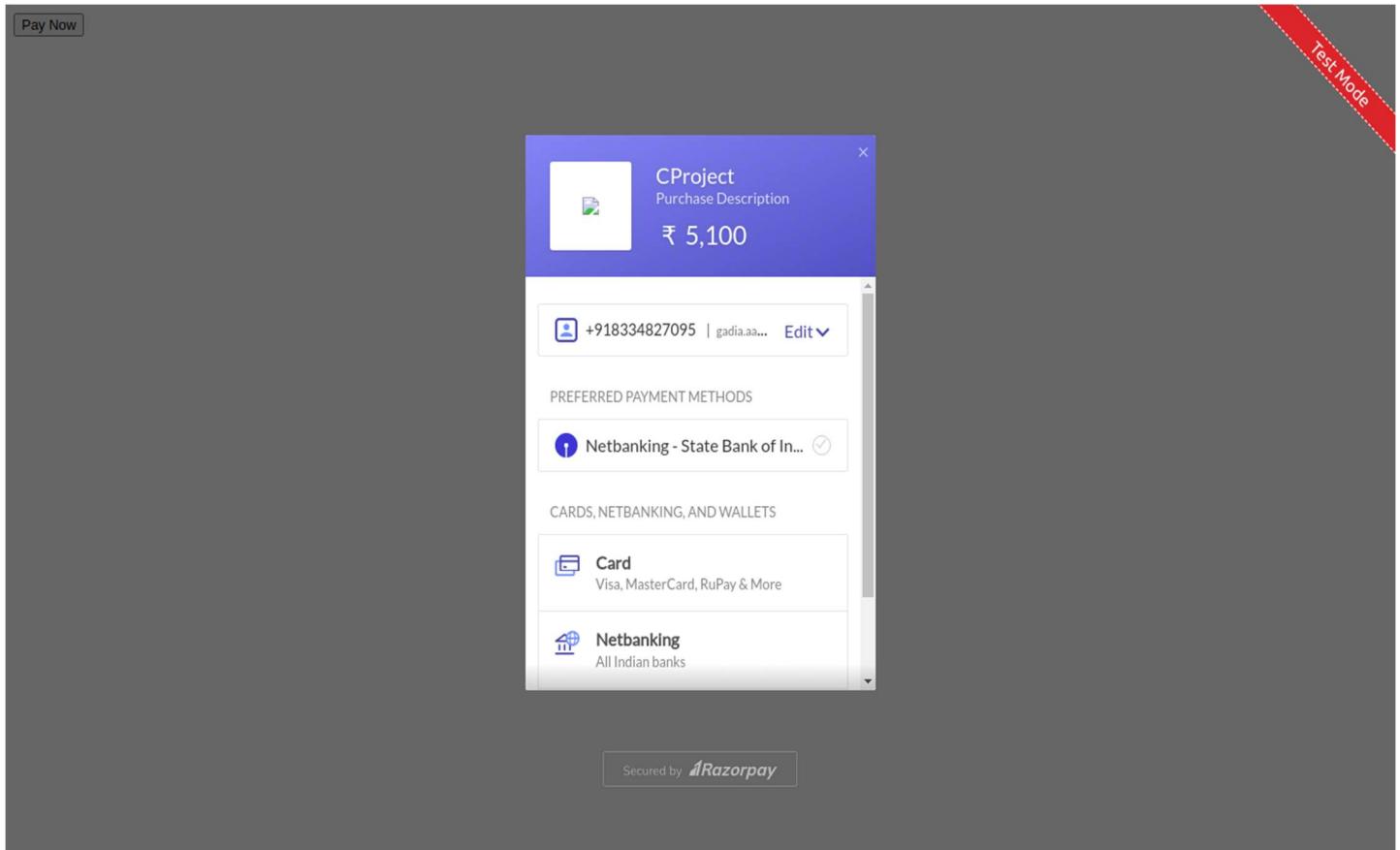
## Fig 9 – User Address Edit Page

**Figure 9**, is used by the **user who has logged-in**, to edit the saved address.

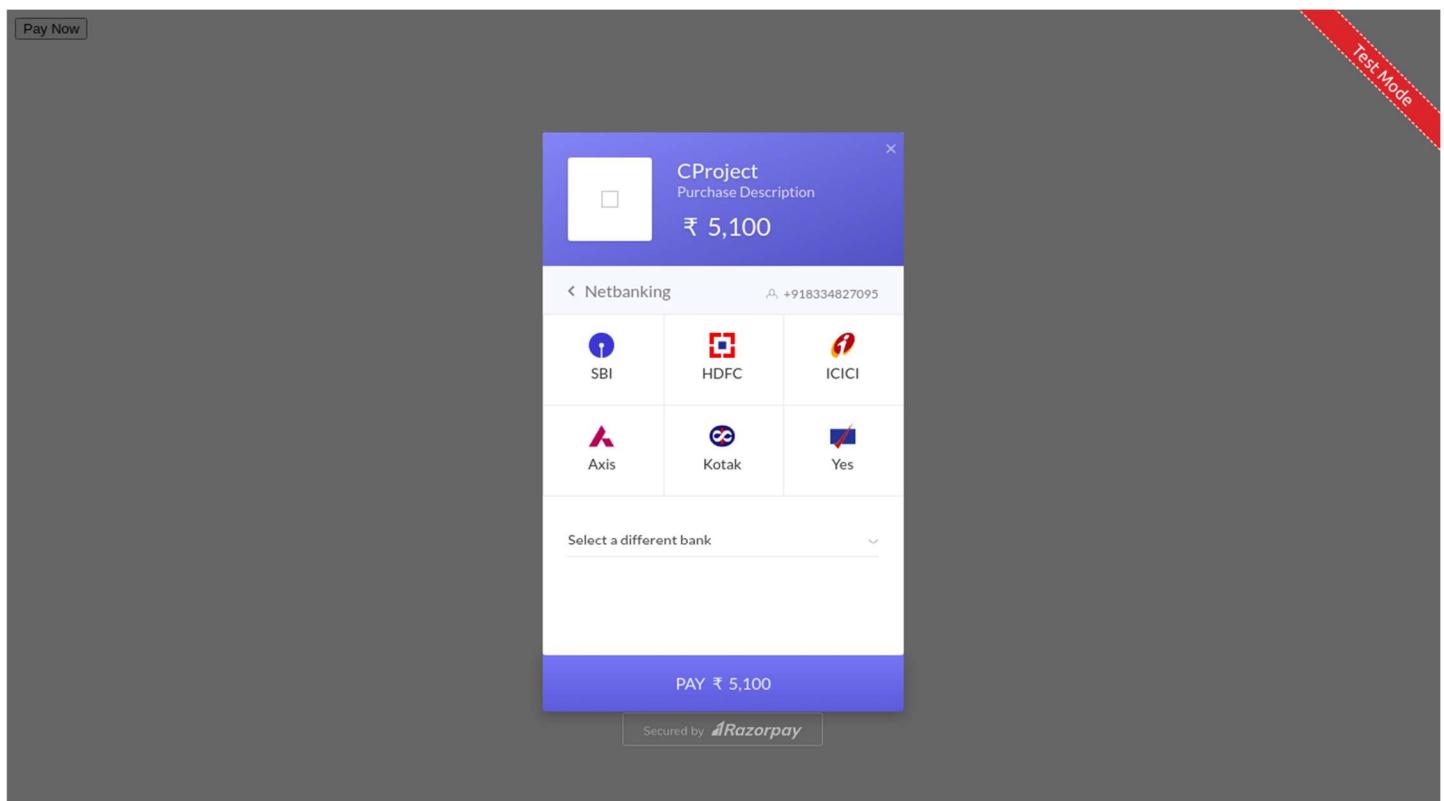


## Fig 10 – Checkout Option

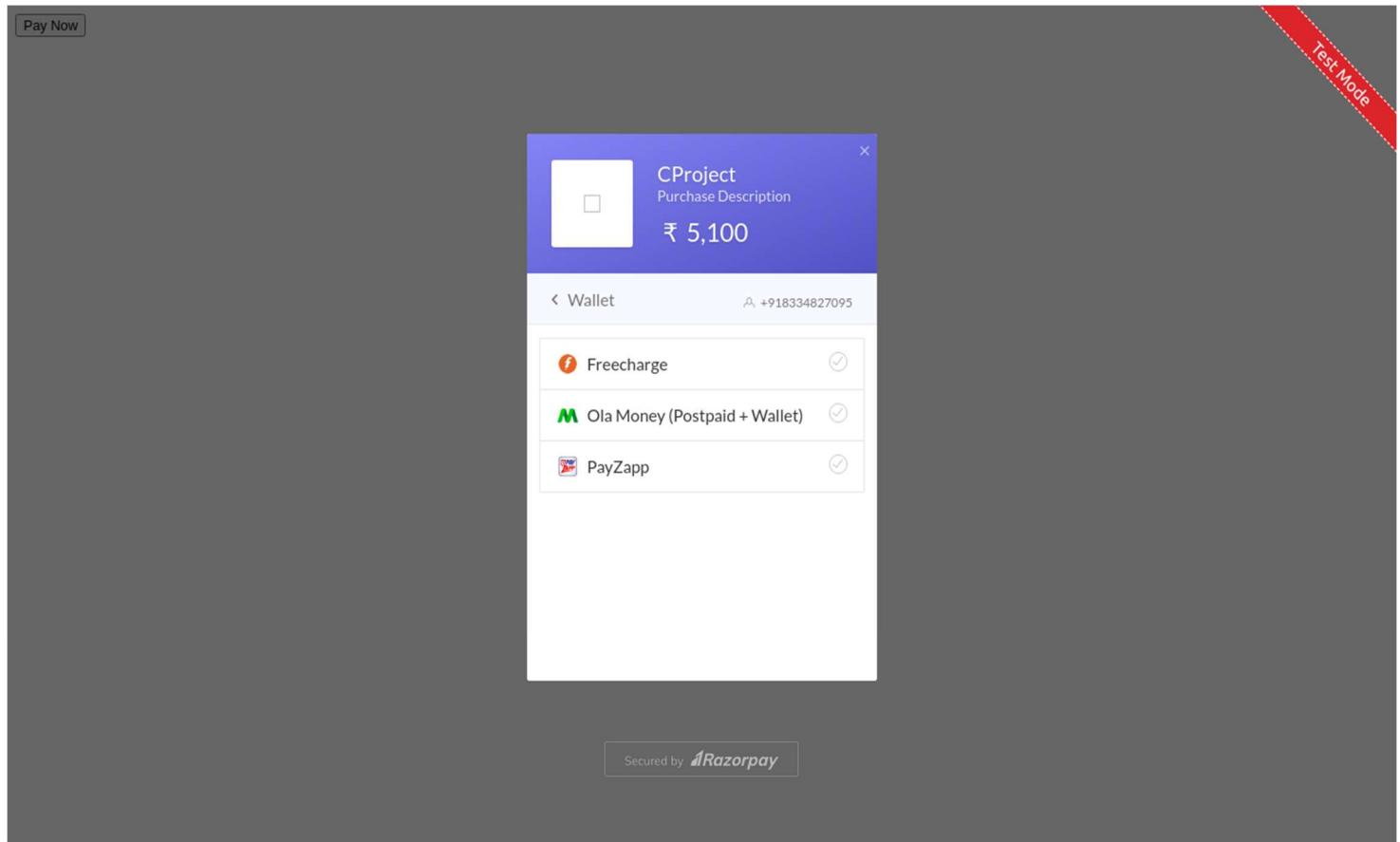
**Figure 10**, is the Checkout Option. User has to click on it to select the payment option.



**Fig 11 – Payment Page-1**



**Fig 12 – Payment Page-2**



**Fig 13 – Payment Page-3**

**Figure 11, 12 & 13,** is how the page looks like once the user clicks on PayNow option. This is the page of the **RazorPay Payment Gateway** which has been used here. User can fill its details, select the payment option & proceed to checkout.

**NOTE for FrontEnd Design-**

**For all the FrontEnd Pages, the report consists of Layout which was designed for each page. All the functionalities have been covered in all the layouts. The codes couldn't be added to the report due to a large number of files.**

**Although the codes though have been presented during the Project Demonstration on 10<sup>th</sup> August' 2020.**

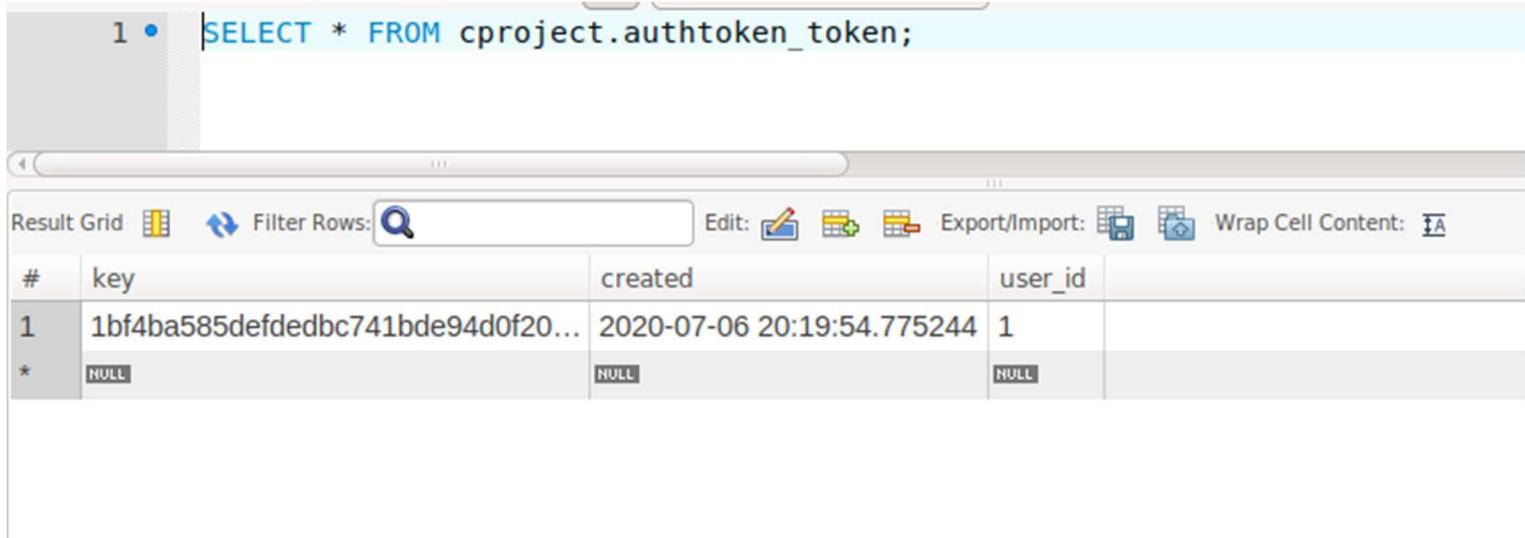
## Database Design

The screenshot shows a database schema browser interface with the title "SCHEMAS" at the top. A search bar labeled "Filter objects" is present. Below it, the "cproject" schema is expanded, showing its contents. The tables listed are:

- auth\_group
- auth\_group\_permissions
- auth\_permission
- authtoken\_token
- auth\_user
- auth\_user\_groups
- auth\_user\_user\_permissions
- bookinventory
- booknotify
- books** (highlighted with a gray background)
- categories
- django\_admin\_log
- django\_content\_type
- django\_migrations
- django\_session
- donationreqs
- faqcategories
- faqs
- new\_pricing\_model
- order\_address
- order\_books
- orders
- pincodes
- pin\_map
- useraddresses
- users

**Fig 14 – Tables in the cproject database**

**Figure 14**, shows the list of **tables** which was created in the cproject database. “**cproject database**”, is the **database used for the website**.



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

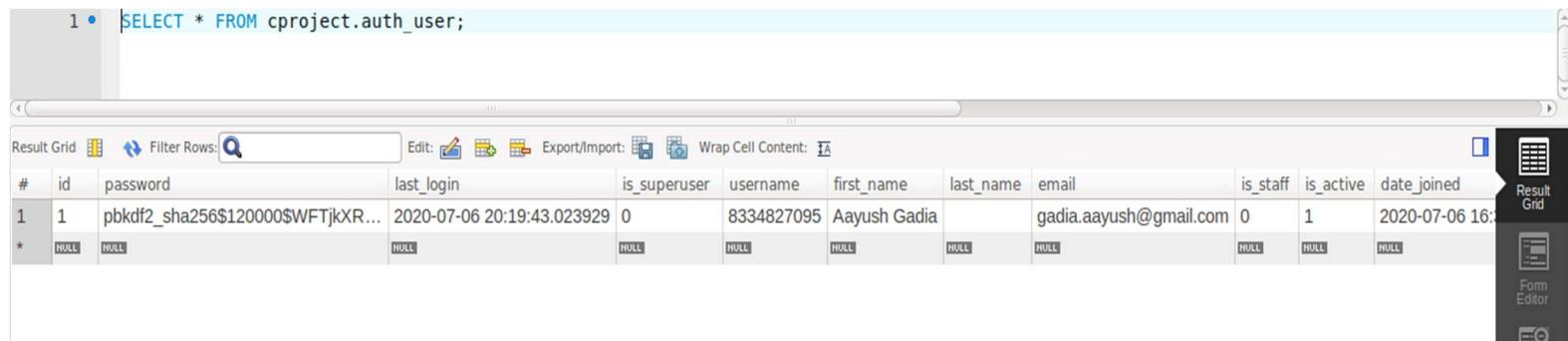
```
1 • |SELECT * FROM cproject.auth_token_token;
```

The result grid displays one row of data:

#	key	created	user_id
1	1bf4ba585defdedbc741bde94d0f20...	2020-07-06 20:19:54.775244	1
*	NULL	NULL	NULL

**Fig 15 – AuthToken Table**

**Figure 15**, table stores the **Token** which has been **allotted to the user once the user has logged in the website**. This **token is verified every time** a user tries to use a **feature on the website** be it to placing orders, to accessing its profile, to cancelling an order and etc.



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 • |SELECT * FROM cproject.auth_user;
```

The result grid displays one row of data:

#	id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined
1	1	pbkdf2_sha256\$12000\$WFTjkXR...	2020-07-06 20:19:43.023929	0	8334827095	Aayush Gadia		gadia.aayush@gmail.com	0	1	2020-07-06 16:19:43.023929
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Fig 16 – AuthUser Table**

**Figure 16**, table **stores the user details after the user signs-up** to the website. All the details user had filled up in the **Sign-up form** are saved here. Passwords are allotted a **hash key using MD5 algorithm** and then it is saved here.

```
1 • | SELECT * FROM cproject.booknotify;
```

#	id	bookid	user_id	book_name	isbn	is_mail_sent	user_email	
1	64012	9277	1	Clark's T...	9788...	N	gadia.aayush@gmail.com	
2	64013	9277	1	Clark's T...	9788...	N	gadia.aayush@gmail.com	
3	64014	9277	1	Clark's T...	9788...	N	gadia.aayush@gmail.com	
4	64015	9277	1	Clark's T...	9788...	N	gadia.aayush@gmail.com	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

**Fig 17 – Booknotify Table**

**Figure 17**, table saves the **details of the Books** which are **out of stock** and the **user who had requested for the notification once it's available**. Basically, when a book is out of stock then the user has the option to click on “Notify-me” on the website. So, when a user clicks on that option then the details are saved here.

```
1 • | SELECT * FROM cproject.books;
```

isbn	title	slug	thumb	author	publication	category	publication_date
9789332518131	Leadership in Organizations (English)	leadership-in-organizations-english-	medium/0802101001459505535.jpg	Gary Yukl	Pearson	333	2015
9788178123936	LIC AAO Exam Guide (English) Lat...	lic-aoe-exam-guide-english-latest-e...	medium/0235415001459507838.jpg	R Gupta	Ramesh	205	2011
9788189973018	Light on the Lotus (English)	light-on-the-lotus-english	medium/0362769001459508620.jpg	Arup Mitra	Readworthy	242	2007
9788172450236	LATEST ESSAYS FOR COLLEGE...	latest-essays-for-college-and-comp...	medium/0144375001459510524.jpg	B. N. AHUJA	Goodwill	205	2013
9780761997528	Leading: Lessons from Literature (...)	leading-lessons-from-literature-engl...	medium/0888850001459510803.jpg	Sampat P Singh	Response Book	333	2003
9789350595992	LITTLE LEADERS	little-leaders-	medium/book_default.jpeg	Shiv Khera	macmillan	272	2014
978063550157	Lots More Tell Me Why	lots-more-tell-me-wh...	medium/0390907001459598393.ind	Arkady Leokum	Dean	263	1994

**Fig 18 – Books Table**

**Figure 18**, table stores the **details of the Books**. **Book information** like it's isbn, title, slug, thumbnail of the image, author, publication, category, publication\_date all are stored here.

1 • |SELECT \* FROM cproject.django\_session;

#	session_key	session_data	expire_date
1	s2hh8w1ve6sr1jrj5e08tmx4rg5hlpah	N2YyOWRiNjM1MjY2YWQyY2FiM...	2020-07-20 20:19:43.400175
*	NULL	NULL	NULL

**Fig 19 – Django Session Table**

**Figure 19**, table stores the **session data** of the **user** who has logged in the website. Let's say a user logged in the website and it **doesn't logs out**, so this is how the **sessions data helps**, basically each user is **given a time period** once it logs in and after that **time period the session expires**, therefore the **user automatically gets logged out**.

1 • |SELECT \* FROM cproject.donationreqs;

#	donation_req_id	user_id	status	Mobile	address	Landmark	country	state	city	zipcode	no_of_book
1	1	1	1	8334827085	test_address_0103	test_0103	test_0103	test0103	test0103	1234560103	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Fig 20 – Donationreqs Table**

**Figure 20**, table stores the **data of the users** who want to **donate their books**. After the **donation form is filled** the data is stored in this table.

```
1 • | SELECT * from cproject.order_address;
```

#	id	order_id	rec_name	email	pincode	address	landmark	city	country	state	phone	courier_name	awb	i_date
1	1	1	aayush_test	NULL	12345	test	test	test	test	test	123456789	NULL	NULL	1594118072
2	2	2	aayush_test	NULL	12345	test	test	test	test	test	123456789	NULL	NULL	1594118167
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Fig 21 – Order Address Table**

**Figure 21,** table stores the **addresses** of the user who have placed the order on the website.

```
1 • | select * from cproject.orders;
```

order_id	b	user_id	amount	no_of_book	status	payusing	checkouttype	billing_add_id	shipping_add_id	i_by	i_date	u_by	u_date	is_thanks_mail_sent	is_notification_sent
1	NULL	1	100	2	2	test_2...	C	1	1	0	1594118072	0	1594118363	N	N
2	NULL	1	100	2	0	test_2...	C	1	1	0	1594118167	0	1594118167	N	N
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Fig 22 – Orders Table**

**Figure 22,** table stores the **order details** of the orders placed by the users of the website. It stores information like the **user\_id**, **amount**, **qty of books ordered**, **mode used for payment**, **date when order placed** and etc.

```
1 •  SELECT * FROM cproject.pincodes;
```

The screenshot shows a MySQL Workbench interface. At the top, a SQL editor window displays the query: "SELECT \* FROM cproject.pincodes;". Below it, a "Result Grid" window shows the data from the table. The grid has three columns: "#", "pincode", and "id". The data consists of 13 rows, each containing a unique ID, a pincode, and a corresponding id value. The "Edit" toolbar at the top right of the grid includes icons for creating, updating, and deleting rows.

#	pincode	id
1	641004	13016
2	641006	13017
3	400601	13018
4	400602	13019
5	390003	13020
6	390007	13021
7	574161	13022
8	121103	13023
9	121107	13024
10	122100	13025
11	122104	13026
12	122107	13027
13	122108	13028

**Fig 23 – Pincodes Table**

**Figure 23,** table stores the different pin codes mapped with a unique id. This will be used when fetching the pincode.

```
1 •  SELECT * FROM cproject.pin_map;
```

The screenshot shows the MySQL Workbench interface with a query editor at the top and a result grid below. The query is 'SELECT \* FROM cproject.pin\_map;'. The result grid has columns: #, pincode, districtname, statename, and id. There are four rows of data:

#	pincode	districtname	statename	id
24	518002	Kurnool	ANDHRA PRADESH	104
25	518001	Kurnool	ANDHRA PRADESH	106
26	518003	Kurnool	ANDHRA PRADESH	107
27	518006	Kurnool	ANDHRA PRADESH	112

**Fig 24 – Pinmap Table**

**Figure 24**, table stores the **mapping of the different pincodes** with their respective **district names, state names & the pincode id** (used in **pincodes table**).

```
1 •  SELECT * FROM cproject.useraddresses;
```

The screenshot shows the MySQL Workbench interface with a query editor at the top and a result grid below. The query is 'SELECT \* FROM cproject.useraddresses;'. The result grid has columns: #, address\_id, user\_id, title, rec\_name, pincode, address, landmark, phone\_no, alt\_phone\_no, address\_type, is\_primary, state\_name, city\_name, and country\_name. There are three rows of data, with the third row partially cut off:

#	address_id	user_id	title	rec_name	pincode	address	landmark	phone_no	alt_phone_no	address_type	is_primary	state_name	city_name	country_name
1	1	1	test_final	aayush_test	12345	test	test	123456789	NULL	0	Y	test	test	test
2	2	1	test_fi...	aayush_t...	123456	test_2	test_2	1234567...	NULL	0	Y	test_2	test_2	test2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Fig 25 – Useraddresses Table**

**Figure 25**, table stores the **addresses** of all the **users**. The table stores the information like **user\_id**, **title**, **name**, **pincode**, **address**, **landmark**, **phone no.** and etc.

The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window with the following content:

```
1 • | SELECT * FROM cproject.categories;
```

Below the code editor is a toolbar with various icons for database management. Underneath the toolbar is a "Result Grid" section. It contains a search bar labeled "Filter Rows:" with a magnifying glass icon. To the right of the search bar are buttons for "Edit" and "Export/Import". The result grid itself has the following structure:

#	id	parent_id	lft	rgt	name	slug
1	187	NULL	381	446	Competitive Exams	competitive-exams
2	190	187	382	395	Engineering & Medical	engineering
3	191	187	396	411	Government Jobs	government-jobs
4	192	187	412	423	International Exams	international-exams
5	193	187	424	435	School Level	school-level
6	194	190	383	384	AIEEE	aieee
7	195	190	385	386	Engineering (Post Graduate)	engineering-post-graduate

**Fig 26 – Categories Table**

**Figure 26**, table stores the **mapping of the different categories of books** with their respective **slugs** , **unique id** & **id of their parent category**.

## Backend API's

Activities Google Chrome Fri 19:40

127.0.0.1:8000/myauth/lo 127.0.0.1:8000/myauth/sig +

← → C ⌂ 127.0.0.1:8000/myauth/signup/

Apps Google Amazon Music

phone [phone] Name [name] Email [email] password [password] Register

Fig 27 – Sign Up in MyAuth API

127.0.0.1:8000/myauth/lo +

← → C ⌂ 127.0.0.1:8000/myauth/login/

Apps Google Amazon Music

**Login**

phone [phone] password [password] Login

Fig 28 – Login in MyAuth API

**Figure 27 & 28**, shows the **Sign-Up & Login API**. Both the API's URL are in the **MyAuth Application**. Also, there is **Form Validation** for all the **data filled in the form**. Both are **POST Method API's**.

All the API's have been built using Django Framework.

Untitled Request

POST http://127.0.0.1:8000/myauth/get-token/

Params Authorization Headers (11) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> phone	8334827095
<input checked="" type="checkbox"/> password	Aayush@2020
Key	Value

Body Cookies (1) Headers (7) Test Results Status:

Pretty Raw Preview Visualize JSON `={`

```
1 {  
2   "code": 201,  
3   "status": "Created",  
4   "message": "Token Created",  
5   "Token": "076b19e8d60fe1e62d094d3d394e7c44361a247d",  
6   "phone": "8334827095"  
7 }
```

**Fig 29 – Token Allotment in MyAuth API**

Untitled Request

GET http://127.0.0.1:8000/myauth/logout/

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Headers `7 hidden`

KEY	VALUE
<input checked="" type="checkbox"/> Authorization	Token ccacf07ed29370f863ad74699359819ebca3a35a
Key	Value

Body Cookies (1) Headers (7) Test Results Status:

Pretty Raw Preview Visualize JSON `={`

```
1 {  
2   "code": 200,  
3   "status": "OK",  
4   "message": "Log Out Successfull"  
5 }
```

**Fig 30 – Logout in MyAuth API**

**Figure 29**, shows the **Token Allotment API**, basically once a user is logged-in then **using this API, a Token is allotted to the User**. This **token is used for accessing all the features on the website like placing an order, placing donation request, viewing orders, cancelling orders, editing profile and etc.** It is a **POST Method API**.

**Figure 30**, shows the **LogOut API**, the **Token allotted to the User is passed as Headers to the API URL**, using which the **log out happen successfully**. After the logout is successful the token is deleted from the database. It is a **GET Method API**.

**Both the API URL's are in the MyAuth Application.**

The screenshot shows a Postman interface with the following details:

- Method:** GET
- URL:** http://127.0.0.1:8000/api/v1/get/get-books/competitive-exams/engineering/iit/15
- Headers:** (selected tab)
- Body:** (radio button selected)
- Params:** none
- Authorization:** none
- Body Content:** This request does not have a body
- Response:**
  - Status: 200 OK
  - Time: 2482 ms
  - Size: 2.49 KB
  - Download
- Body Content (JSON):**

```
[{"book_id": 39235, "title": "DEEKSHA ASSIGNMENT MANUAL FOR IIT JEE PACK OF 16 BOOKS", "thumb": "0668105001535546333.jpg", "price": 8000.0, "slug": "deeksha-assignment-manual-for-iit-jee-pack-of-16-books"}, {"book_id": 39178, "title": "BRILLIANT TUTORIALS IIT JEE ELITE-2006 WITH YG-FILE SCREENING AND MAIN IIT-JEE 2005 PACK OF 15 BOOKS", "thumb": "0380065001535527608.jpg", "price": 5000.0, "slug": "brilliant-tutorials-iit-jee-elite-2006-with-yg-file-screening-and-main-iit-jee-2005-pack-of-15-books"}, {"book_id": 39170, "title": "BRILLIANT TUTORIALS YG-FILE FOR IIT-JEE PACK OF 4 BOOKS IN ENGLISH", "thumb": "0758424001535525978.jpg", "price": 2000.0, "slug": "brilliant-tutorials-yg-file-for-iit-jee-pack-of-4-books-in-english"}, {"book_id": 39103, "title": "VIDYALANKAR STUDYMATERIAL FOR IIT-JEE,CLASS 11 AND CLASS 12 WITH SOLUTION TO JEE-MAIN EXAMINATION PACK OF 21 BOOKS", "thumb": "0245978001535445470.jpg", "price": 8000.0, "slug": "0447139001535450035.jpg"}, {"book_id": 39087, "title": "RAO IIT ACADEMY STUDY MATERIAL PACK OF 53 BOOKS", "thumb": "0447139001535450035.jpg", "price": 8000.0, "slug": "rao-iit-academy-study-material-pack-of-53-books"}, {"book_id": 39042, "title": "BASE IIT JEE ACHIEVER FOR 2 STUDENTS PACK OF 13 BOOKS", "thumb": "0830792001535436143.jpg", "price": 6000.0, "slug": "base-iit-jee-achiever-for-2-students-pack-of-13-books"}, {"book_id": 38987, "title": "NARAYANA IIT/PMT ACADEMY STUDY MATERIAL FOR CLASS 11 & 12,JEE-MAIN AND JEE-ADVANCED PACK OF 81 BOOKS", "thumb": "0978828001535362884.jpg", "price": 25000.0, "slug": "narayana-iitpmt-academy-study-material-for-class-11jee-main-and-jee-advanced-pack-of-81-books"}, {"book_id": 38974, "title": "NARAYANA STUDY MATERIAL FOR 11,12 AND JEE ADVANCED PACK OF 16 BOOKS", "thumb": "book_default.jpeg", "price": 6000.0, "slug": "narayana-study-material-for-11,12-and-jee-advanced-pack-of-16-books"}, {"book_id": 38964, "title": "STUDY VAULT STUDY MATERIAL PACK OF 28 BOOKS", "thumb": "0375772001535355785.jpg", "price": 12000.0, "slug": "study-vault-study-material-pack-of-28-books"}, {"book_id": 38959, "title": "PACE STUDY MATERIAL FOR IIT PACK OF 64 BOOKS", "thumb": "0179722001535352982.jpg", "price": 25000.0, "slug": "pace-study-material-for-iit-pack-of-64-books"}]
```

**Fig 31 – Get-books URL - 1, GetBooks API**

GET http://127.0.0.1:8000/api/v1/get/category/competitive-exams/engineering/iit/15/ Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 2137 ms Size: 2.49 KB Download

Pretty Raw Preview HTML

```
1 [{"book_id": 39235, "title": "DEEKSHA ASSIGNMENT MANUAL FOR IIT JEE PACK OF 16 BOOKS", "thumb": "0668105001535546333.jpg", "price": 8000.0, "slug": "deeksha-assignment-manual-for-iit-jee-pack-of-16-books"}, {"book_id": 39178, "title": "BRILLIANT TUTORIALS IIT JEE ELITE-2006 WITH YG-FILE SCREENING AND MAIN IIT-JEE 2005 PACK OF 15 BOOKS", "thumb": "0380065001535527608.jpg", "price": 5000.0, "slug": "brilliant-tutorials-iit-jee-elite-2006-with-yg-file-screening-and-main-iit-jee-2005-pack-of-15-books"}, {"book_id": 39170, "title": "BRILLIANT TUTORIALS YG-FILE FOR IIT-JEE PACK OF 4 BOOKS IN ENGLISH", "thumb": "0758424001535525978.jpg", "price": 2000.0, "slug": "brilliant-tutorials-yg-file-for-iit-jee-pack-of-4-books-in-english"}, {"book_id": 39103, "title": "VIDYALANKAR STUDYMATERIAL FOR IIT-JEE,CLASS 11 AND CLASS 12 WITH SOLUTION TO JEE-MAIN EXAMINATION PACK OF 21 BOOKS", "thumb": "0447139001535450035.jpg", "price": 8000.0, "slug": "vidyalankar-studymaterial-for-iit-jeclass-11-and-class-12-with-solution-to-jee-main-examination-pack-of-21-books"}, {"book_id": 39087, "title": "RAO IIT ACADEMY STUDY MATERIAL PACK OF 53 BOOKS", "thumb": "02459780015354545470.jpg", "price": 25000.0, "slug": "rao-iit-academy-study-material-pack-of-53-books"}, {"book_id": 39042, "title": "BASE IIT JEE ACHIEVER FOR +2 STUDENTS PACK OF 13 BOOKS", "thumb": "0830792001535436143.jpg", "price": 6000.0, "slug": "base-iit-jee-achiever-for-2-students-pack-of-13-books"}, {"book_id": 38987, "title": "NARAYANA IIT/PMT ACADEMY STUDY MATERIAL FOR CLASS 11,12,JEE-MAIN AND JEE-ADVANCED PACK OF 81 BOOKS", "thumb": "0978828001535362884.jpg", "price": 25000.0, "slug": "narayana-iitpmt-academy-study-material-for-class-1112jee-main-and-jee-advanced-pack-of-81-books"}, {"book_id": 38974, "title": "NARAYANA STUDY MATERIAL FOR 11,12 AND JEE ADVANCED PACK OF 16 BOOKS", "thumb": "book-default.jpeg", "price": 6000.0, "slug": "narayana-study-material-for-1112-and-jee-advanced-pack-of-16-books"}, {"book_id": 38964, "title": "STUDY VAULT STUDY MATERIAL PACK OF 28 BOOKS", "thumb": "0375772001535355785.jpg", "price": 12000.0, "slug": "study-vault-study-material-pack-of-28-books"}, {"book_id": 38959, "title": "PACE STUDY MATERIAL FOR IIT PACK OF 64 BOOKS", "thumb": "0179722001535352902.jpg", "price": 25000.0, "slug": "pace-study-material-for-iit-pack-of-64-books"}]
```

**Fig 32 – Category URL - 1, GetBooks API**

GET http://127.0.0.1:8000/api/v1/get/get-books/810/ Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 2256 ms Size: 2.01 KB Download

Pretty Raw Preview HTML

```
1 [{"book_id": 50712, "title": "Learn the art of drawing and shading By Manoj jain", "thumb": "0400990001542609009.jpg", "price": 40.0, "slug": "learn-the-art-of-drawing-and-shading-by-manoj-jain"}, {"book_id": 50711, "title": "Chhota Bheem Flying Carpet By Rajiv Chilaka", "thumb": "0694053001542608961.jpg", "price": 70.0, "slug": "chhota-bheem-flying-carpet-by-rajiv-chilaka"}, {"book_id": 50710, "title": "Scream for Ice Cream (Nancy Drew and the Clue Crew Book 2) By Carolyn Keene", "thumb": "0544445001542608886.jpg", "price": 250.0, "slug": "scream-for-ice-cream-nancy-drew-and-the-clue-crew-book-2-by-carolyn-keene"}, {"book_id": 50709, "title": "Drama in real life By NA", "thumb": "0031365001542608773.jpg", "price": 20.0, "slug": "drama-in-real-life-by-na"}, {"book_id": 50708, "title": "Journey To The Centre Of The Earth By Jules Verne", "thumb": "0148627001542608705.jpg", "price": 100.0, "slug": "journey-to-the-centre-of-the-earth-by-jules-verne"}, {"book_id": 50707, "title": "Chinai 2010 By NA", "thumb": "0431881001542608767.jpg", "price": 150.0, "slug": "chinai-2010-by-na"}, {"book_id": 50706, "title": "Chhota Bheem Vol. 32 By Rajiv Chilaka", "thumb": "0724642001542608745.jpg", "price": 99.0, "slug": "chhota-bheem-vol-32-by-rajiv-chilaka"}, {"book_id": 50705, "title": "Calendar Mysteries #3: March Mischief (A Stepping Stone Book(TM)) By Ron Roy", "thumb": "0634468001542608567.jpg", "price": 95.0, "slug": "calendar-mysteries-3-march-mischief-a-stepping-stone-booktm-by-ron-roy"}, {"book_id": 50704, "title": "The Mum-Minder By Jacqueline Wilson", "thumb": "0132753001542608521.jpg", "price": 250.0, "slug": "the-mum-minder-by-jacqueline-wilson"}, {"book_id": 50703, "title": "Tell Me Why Ideas That Change The World By na", "thumb": "0667013001542608578.jpg", "price": 20.0, "slug": "tell-me-why-ideas-that-change-the-world-by-na"}]
```

**Fig 33 – Get-books URL - 2, GetBooks API**

GET http://127.0.0.1:8000/api/v1/get/category/810/ Send Save

Params Authorization Headers **Body** Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary

This request does not have a body

**Body** Cookies (1) Headers (6) Test Results Status: 200 OK Time: 2834 ms Size: 2.01 KB Download

Pretty Raw Preview HTML

```
i 1 [{"book_id": 50712, "title": "Learn the art of drawing and shading By Manoj jain", "thumb": "0400990001542609009.jpg", "price": 40.0, "slug": "learn-the-art-of-drawing-and-shading-by-manoj-jain"}, {"book_id": 50711, "title": "Chhota Bheem Flying Carpet By Rajiv Chilaka", "thumb": "0694053001542608961.jpg", "price": 70.0, "slug": "chhota-bheem-flying-carpet-by-rajiv-chilaka"}, {"book_id": 50710, "title": "Scream for Ice Cream (Nancy Drew and the Clue Crew Book 2) By Carolyn Keene", "thumb": "0544445001542608886.jpg", "price": 250.0, "slug": "scream-for-ice-cream-nancy-drew-and-the-clue-crew-book-2-by-carolyn-keene"}, {"book_id": 50709, "title": "Drama in real life By NA", "thumb": "0031365001542608773.jpg", "price": 20.0, "slug": "drama-in-real-life-by-na"}, {"book_id": 50708, "title": "Journey To The Centre Of The Earth By Jules Verne", "thumb": "0148627001542608705.jpg", "price": 100.0, "slug": "journey-to-the-centre-of-the-earth-by-jules-verne"}, {"book_id": 50707, "title": "Chinal 2010 By NA", "thumb": "0431881001542608767.jpg", "price": 150.0, "slug": "chinai-2010-by-na"}, {"book_id": 50706, "title": "Chhota Bheem Vol. 32 By Rajiv Chilaka", "thumb": "0724642001542608745.jpg", "price": 99.0, "slug": "chhota-bheem-vol-32-by-rajiv-chilaka"}, {"book_id": 50705, "title": "Calendar Mysteries #3: March Mischief (A Stepping Stone Book(TM)) By Ron Roy", "thumb": "0634468001542608567.jpg", "price": 95.0, "slug": "calendar-mysteries-3-march-mischief-a-stepping-stone-booktm-by-ron-roy"}, {"book_id": 50704, "title": "The Mum-Minder By Jacqueline Wilson", "thumb": "0132753001542608521.jpg", "price": 250.0, "slug": "the-mum-minder-by-jacqueline-wilson"}, {"book_id": 50703, "title": "Tell Me Why Ideas That Change The World By na", "thumb": "0667013001542608578.jpg", "price": 20.0, "slug": "tell-me-why-ideas-that-change-the-world-by-na"}]
```

**Fig 34 – Category URL - 2, GetBooks API**

GET http://127.0.0.1:8000/api/v1/get/product/810/ Send Save

Params Authorization Headers (1) **Body** Pre-request Script Tests Cookies Code Comments (0)

none form-data x-www-form-urlencoded raw binary

This request does not have a body

**Body** Cookies (1) Headers (6) Test Results Status: 200 OK Time: 4368 ms Size: 2.01 KB Download

Pretty Raw Preview HTML

```
i 1 [{"book_id": 50712, "title": "Learn the art of drawing and shading By Manoj jain", "thumb": "0400990001542609009.jpg", "price": 40.0, "slug": "learn-the-art-of-drawing-and-shading-by-manoj-jain"}, {"book_id": 50711, "title": "Chhota Bheem Flying Carpet By Rajiv Chilaka", "thumb": "0694053001542608961.jpg", "price": 70.0, "slug": "chhota-bheem-flying-carpet-by-rajiv-chilaka"}, {"book_id": 50710, "title": "Scream for Ice Cream (Nancy Drew and the Clue Crew Book 2) By Carolyn Keene", "thumb": "0544445001542608886.jpg", "price": 250.0, "slug": "scream-for-ice-cream-nancy-drew-and-the-clue-crew-book-2-by-carolyn-keene"}, {"book_id": 50709, "title": "Drama in real life By NA", "thumb": "0031365001542608773.jpg", "price": 20.0, "slug": "drama-in-real-life-by-na"}, {"book_id": 50708, "title": "Journey To The Centre Of The Earth By Jules Verne", "thumb": "0148627001542608705.jpg", "price": 100.0, "slug": "journey-to-the-centre-of-the-earth-by-jules-verne"}, {"book_id": 50707, "title": "Chinal 2010 By NA", "thumb": "0431881001542608767.jpg", "price": 150.0, "slug": "chinai-2010-by-na"}, {"book_id": 50706, "title": "Chhota Bheem Vol. 32 By Rajiv Chilaka", "thumb": "0724642001542608745.jpg", "price": 99.0, "slug": "chhota-bheem-vol-32-by-rajiv-chilaka"}, {"book_id": 50705, "title": "Calendar Mysteries #3: March Mischief (A Stepping Stone Book(TM)) By Ron Roy", "thumb": "0634468001542608567.jpg", "price": 95.0, "slug": "calendar-mysteries-3-march-mischief-a-stepping-stone-booktm-by-ron-roy"}, {"book_id": 50704, "title": "The Mum-Minder By Jacqueline Wilson", "thumb": "0132753001542608521.jpg", "price": 250.0, "slug": "the-mum-minder-by-jacqueline-wilson"}, {"book_id": 50703, "title": "Tell Me Why Ideas That Change The World By na", "thumb": "0667013001542608578.jpg", "price": 20.0, "slug": "tell-me-why-ideas-that-change-the-world-by-na"}]
```

**Fig 35 – Product URL, GetBooks API**

**Figure 31 - 35**, shows the **GetBooks Application**. It basically fetches the details of the books like the **book\_id**, **title**, **publisher**, **author**, **publishing date**, **slug**, **price**, **no\_of\_pages** and etc.

These information are used to show the information on the Home-Page and the Categories Page. It is a **GET Method API**.

There are different **URL's for the same functionality in the Application**. All these are shown above. These different URL's give the users a flexibility, therefore, a user could view a page in different ways. Also, this improves the ranking of pages in the search indexing.

The screenshot shows a Postman request for the URL `http://127.0.0.1:8000/api/v1/get/product/object-oriented-programming-in-turbo-c-in-english`. The request method is GET. The Headers tab is selected, showing two entries: 'Content-Type' and 'Accept'. The Body tab is selected, showing the response body. The response status is 200 OK, with a time of 9462 ms and a size of 1.41 KB. The response body is a JSON object:

```
1 {"product_details": {"book_id": 18159, "title": "OBJECT ORIENTED PROGRAMMING IN TURBO C++ IN ENGLISH", "thumb": "0554561001498913719.jpg", "price": 279.0, "slug": "object-oriented-programming-in-turbo-c-in-english", "publication": "GALGOTIA", "publication_date": 2000, "language": "English", "binding": "paper", "weight": "1.4", "no_of_pages": 739, "edition": null, "isbn": "8185623228", "author": "ROBERT LAFORE", "book_desc": ""}, "product_condition": {"AverageButInReadableCondition": {"shipping": 138.0, "book_inv_id": 44276, "rack_no": "KOL/35/3/57", "donor_name": "Sreenivasa Kumar Hari", "avatar": "0595330001505550181.jpg", "total_qty": 3, "book_inv_id's": [44276, 44279, 44282]}, "BrandNew": {"shipping": "NA", "book_inv_id": 68229, "rack_no": "KOL/76/5/68", "donor_name": "Vipin Gaur", "avatar": null, "total_qty": 1, "book_inv_id's": [68229]}, "VeryGood": {"shipping": "NA", "book_inv_id": 44277, "rack_no": "KOL/41/1/38", "donor_name": "Subramani", "avatar": null, "total_qty": 8, "book_inv_id's": [44277, 44278, 44280, 44284, 44285, 49858, 50696, 62748]}, "AlmostNew": {"shipping": "NA", "book_inv_id": 68004, "rack_no": "KOL/76/5/13", "donor_name": "Bharpreet Singh", "avatar": null, "total_qty": 1, "book_inv_id's": [68004]}}]
```

**Fig 36 – GetProductInfo API**

**Figure 36**, shows the **GetProductInfo Application**. This basically fetches the information of a book for the product page. Information like **book\_id**, **title**, **thumbnail**, **price**, **binding**, **publication**, **publication\_date**, **condition of books** and etc. are returned. It is a **GET Method API**.

This information is used for the **Product page of the books**.

The screenshot shows a POST request to `http://127.0.0.1:8000/api/v1/post/notify_me`. The request body is a JSON object:

```
1 {  
2   "data":  
3     {  
4       "slug": "clarks-tables-science-data-book"  
5     }  
6 }
```

The response status is `201 Created`, time `2907 ms`, size `334 B`. The response body is:

```
1 {  
2   "status": 200,  
3   "message": "Notification of Book Title- Clark's Tables Science Data Book , added successfully."  
4 }
```

**Fig 37 – BookNotify API - 1**

The screenshot shows a POST request to `http://127.0.0.1:8000/api/v1/post/notify_me`. The request body is a JSON object with an empty `slug` field:

```
1 {  
2   "data":  
3     {  
4       "slug": ""  
5     }  
6 }
```

The response status is `400 Bad Request`, time `1144 ms`, size `276 B`. The response body is:

```
1 {  
2   "status": 400,  
3   "message": "failure- check readme"  
4 }
```

**Fig 38 – BookNotify API - 2**

**Figure 37 & 38**, shows the **Booknotify Application**. Therefore whenever a user clicks on the “Notify Me” option on the page, **then this is the API which gets executed**. The API stores the **information of the book & the user who requested for the book**. It is a **POST Method API**.

The API works only when the **user is logged-in**. Therefore, **token has to be passed in the Headers**.

The screenshot shows a Postman interface for a POST request to `http://127.0.0.1:8000/api/v1/post/cancel_order`. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "data" :  
3     {  
4       "order_id" : 521153  
5     }  
6 }
```

The response section shows a status of 200 OK with the message "Cancellation Successful".

Body	Cookies (1)	Headers (7)	Test Results	Status: 200 OK	Time: 4290 ms	Size: 269 B	Download
Pretty	Raw	Preview	JSON				

**Fig 39 – Order Cancel API**

**Figure 39**, shows the **Order Cancel Application**. This is **used to cancel the orders which has been placed**. It is a **POST Method API**.

The API works only when the **user is logged-in**. Therefore, **token has to be passed in the Headers**.

POST http://127.0.0.1:8000/api/v1/post/order\_details

**Body**

```

1 { "data" :
2   {
3     "amount" : 100,
4     "no_of_book" : 2,
5     "payusing" : "test_0103",
6     "billing_add_id" : 32921,
7     "shipping_add_id" : 32921,
8     "fast_delivery" : 0,
9     "shipping_handling_donation" : 10,
10    "prepaid_discount" : 30,
11    "delivery_charge_newbooks" : 100,
12    "coupon_id" : "MP10",
13    "coupon_amount" : 10,
14    "wallet_used" : 10
15  }
16 }
17

```

**Body** Cookies (1) Headers (7) Test Results Status: 201 Created Time: 1850 ms Size: 318 B Download

Pretty Raw Preview JSON

```

1 {
2   "status": 200,
3   "message": "Order Details of Order booked by User Id- 214 , added successfully."
4 }

```

**Fig 40 – Order Details API**

GET http://127.0.0.1:8000/api/v1/get/user\_orders/1

**Headers (2)**

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
Authorization	Token da0a3bed7fd86b67f0cddd7f49248813a14f00f4				
Content-Type	application/json				
Key	Value	Description			

**Body** Cookies (1) Headers (7) Test Results Status: 200 OK Time: 972 ms Size: 479 B Download

Pretty Raw Preview JSON

```

1 {
2   "status": 200,
3   "message": "success",
4   "output": [
5     {
6       "order_id": 521638,
7       "amount": 100,
8       "no_of_book": 1,
9       "actual_date_upload": null,
10      "status": 0,
11      "payusing": "test",
12      "i_date": 123456789,
13      "title": "aayush_test",
14      "thumb": "test.jpg",
15      "price": 1
16    }
17  ]
18 }

```

**Fig 41 – User Order Fetch API**

**Figure 40**, shows the **Order Details Application**. Basically, once a **user places an order**, then **this application is used to send the Order Details to the database**. Using this application the **order details are saved to the database**. It is a **POST Method API**.

**Figure 41**, shows the **User Order Fetch Application**. Basically, when a **user tries to view the orders it has placed on the website**, then, **this application is used**. It **fetches all the orders that user has placed, with the more recent orders being shown first**. It is **GET Method API**.

Both the API works only when the **user is logged-in**. Therefore, **token has to be passed in the Headers**.

The screenshot shows a POST request in Postman. The URL is `http://127.0.0.1:8000/api/v1/post/address_form`. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "data":  
3     {  
4       "title": "test_final_2",  
5       "rec_name": "aayush_test",  
6       "pincode": 12345,  
7       "address": "test",  
8       "landmark": "test",  
9       "phone_no": 123456789,  
10      "state_name": "test",  
11      "city_name": "test",  
12      "country_name": "test",  
13      "is_primary": "Y"  
14    }  
15 }
```

The response status is 201 Created, time: 1250 ms, size: 296 B. The response body is:

```
1 {  
2   "status": 200,  
3   "message": "Address of User Id- 214 , added successfully."  
4 }
```

Fig 42 – User Address Add API

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8000/api/v1/post/edit\_address
- Body (JSON):**

```

1 {
2   "data" :
3     {
4       "address_id" : 32921,
5       "title" : "test_final_2_update",
6       "rec_name" : "aayush_test_2",
7       "pincode" : 123456,
8       "address" : "test_2",
9       "landmark" : "test_2",
10      "phone_no" : 1234567892,
11      "state_name" : "test_2",
12      "city_name" : "test_2",
13      "country_name" : "test2",
14      "is_primary" : "y"
15    }
16  }
17

```
- Response Status:** 201 Created
- Response Time:** 1318 ms
- Response Size:** 298 B

**Fig 43 – User Address Edit API**

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8000/api/v1/post/delete\_address
- Body (JSON):**

```

1 {
2   "address_id":33010
3 }

```
- Response Status:** 202 Accepted
- Response Time:** 556 ms
- Response Size:** 267 B

**Fig 44 – User Address Delete API**

The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8000/api/v1/get/user_address`. The Headers tab is selected, containing `Content-Type: application/json` and `Authorization: Token da0a3bed7fd86b67f0cddd7f49248813a14f00f4`. The Body tab shows a JSON response with two address objects:

```

1  [{"status":200,"message":"success","output":[{"address_id":32913,"user_id":214,"title":"test_2702","rec_name":"test_aayush","pincode":700156,"address":"test","landmark":"test","country_id":1,"state_id":0,"city_id":0,"phone_no":8334827085,"address_type":0,"is_primary":N,"state_name":west bengal,"city_name":kolkata,"country_name":india}, {"address_id":32918,"user_id":214,"title":"test_2702","rec_name":"test_aayush","pincode":700156,"address":test,"landmark":test,"country_id":1,"state_id":0,"city_id":0,"phone_no":8334827085,"address_type":0,"is_primary":N,"state_name":west bengal,"city_name":kolkata,"country_name":india}]}

```

**Fig 45 – User Address Fetch API**

**Figure 42**, shows the **User Address Add API**. Basically, this **application is used by the user to add & save address to its profile on the website**. This helps the user to use the saved address during checkout , thereby, preventing them from entering the details repeatedly.

Similarly, **Figure 43**, shows the **User Address Edit API**. This is used to **edit the address which user had saved previously**.

**Figure 44**, shows the **User Address Delete API**, this is used to **delete the address which user has saved on the website**.

All the above three are **POST Method API's**.

Finally, **Figure 45**, shows the **User Address Fetch API**, it is used to **fetch all the addresses that user has saved on the website**. It is a **GET method API**.

All the Four API's works only when the **user is logged-in**. Therefore, token has to be passed in the Headers.

The screenshot shows a POST request to `http://127.0.0.1:8000/api/v1/post/get_bookinfo`. The **Body** tab is selected, displaying the following JSON payload:

```
1 { "data" :  
2 {  
3 "isbn" : 9789610104841  
4 }  
5 }
```

The response body is displayed under the **Body** tab, showing a detailed JSON object representing a book volume:

```
i 1 {  
2 "kind": "books#volumes",  
3 "totalItems": 1,  
4 "items": [  
5 {  
6 "kind": "books#volume",  
7 "id": "u7ikks6JkRYC",  
8 "etag": "8Jbqtx5Vm6g",  
9 "selfLink": "https://www.googleapis.com/books/v1/volumes/u7ikks6JkRYC",  
10 "volumeInfo": {  
11 "title": "Tisoč veličastnih sonc",  
12 "authors": [  
13 "Khaled Hosseini"],  
14 },  
15 "publisher": "MLADINSKA KNJIGA ZALOZBA",  
16 "publishedDate": "2008",  
17 "industryIdentifiers": [  
18 {  
19 "type": "ISBN_13",  
20 "identifier": "9789610104841"  
21 },
```

**Fig 46 – GetBookInfo API**

**Figure 46**, shows the **GetBookInfo API**. Basically, this API fetches the information on books like title, id, authors, publication, no. of pages, price and etc.

This API works on top of Google Books API. It is a POST Method API.

← → C ⌂ ① 127.0.0.1:8000/search/physics/1

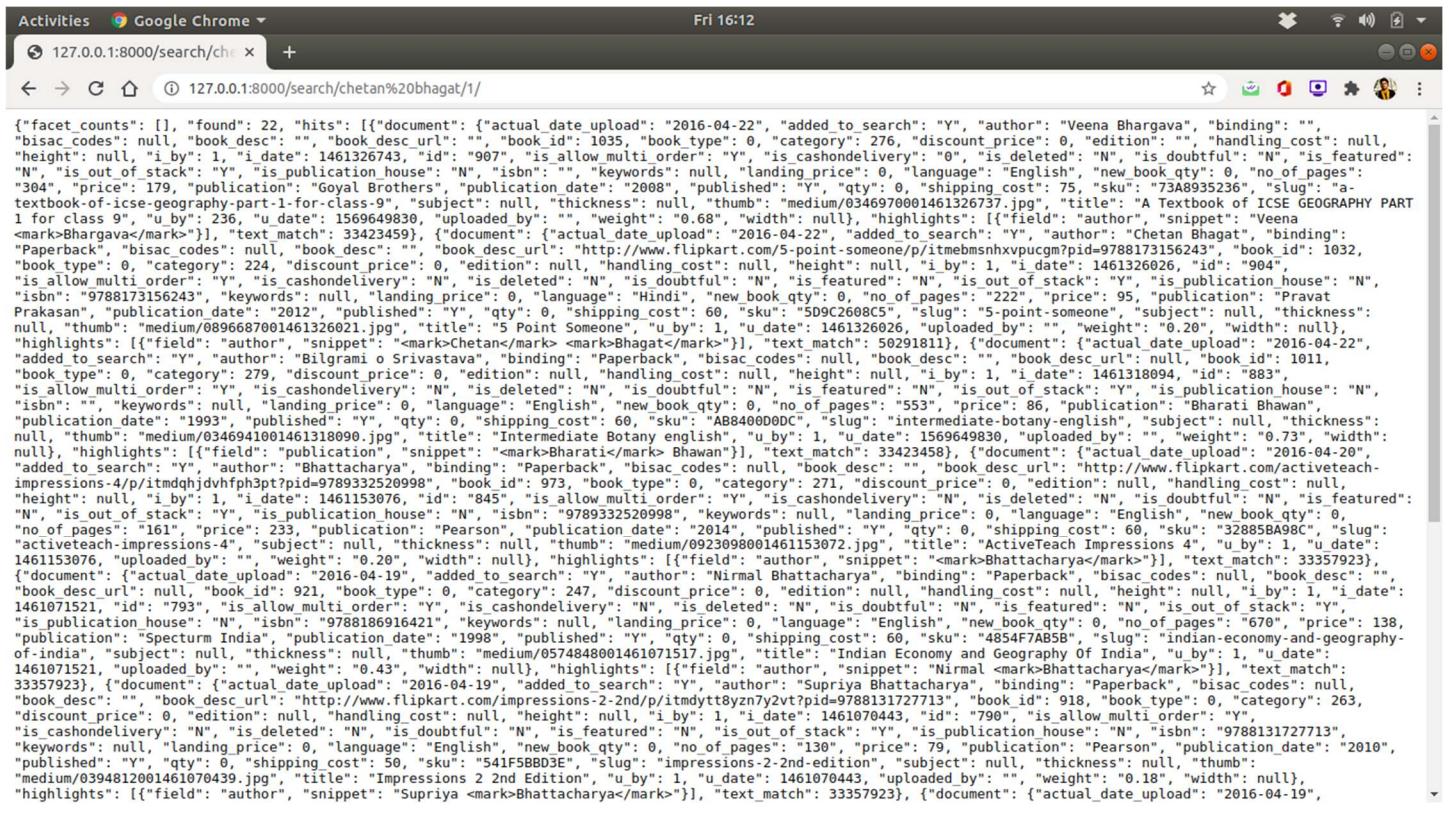
```
{
  "facet_counts": [],
  "found": 32,
  "hits": [
    {
      "document": {
        "actual_date_upload": "2016-04-23",
        "added_to_search": "Y",
        "author": "K kMohindroo",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 1118,
        "book_type": 0,
        "category": 278,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461406032,
        "id": "990",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_deleted": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": "250",
        "price": 130,
        "publication": "Ptambar",
        "publication_date": "2008",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 60,
        "sku": "E6697F1965",
        "slug": "isc-practical-physics-vol-1-for-class-11",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0248867001461406029.jpg",
        "title": "ISC practical Physics vol 1 for class 11",
        "u_by": 1,
        "u_date": 1569649830,
        "uploaded_by": "",
        "weight": "0.25",
        "width": null,
        "highlights": [
          {
            "field": "title",
            "snippet": "ISC practical Physics vol 1 for class 11"
          }
        ],
        "text_match": "33488996"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-23",
        "added_to_search": "Y",
        "author": "Na",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": "http://www.bookadda.com/books/physics-class-12-with-xtra-innings-20-20-cbse-233512",
        "book_id": 1075,
        "book_type": 0,
        "category": 279,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461392393,
        "id": "947",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "N",
        "is_publication_house": "N",
        "isbn": "97888181372321",
        "publication_date": "2008",
        "published": "Y",
        "qty": 1,
        "shipping_cost": 60,
        "sku": "0E536325FF",
        "slug": "physics-with-xtra-innings-class-12",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0685628001461392389.jpg",
        "title": "Physics with Xtra innings class 12",
        "u_by": 1,
        "u_date": 1573815859,
        "uploaded_by": "",
        "weight": "0.58",
        "width": null,
        "highlights": [
          {
            "field": "title",
            "snippet": "Physics with Xtra innings class 12"
          }
        ],
        "text_match": "33488996"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-23",
        "added_to_search": "Y",
        "author": "R K Gaur",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 1066,
        "book_type": 0,
        "category": 313,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461390357,
        "id": "938",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": "371",
        "price": 152,
        "publication": "Rachana Sagar",
        "publication_date": "2008",
        "published": "Y",
        "qty": 1,
        "shipping_cost": 60,
        "sku": "1569649830",
        "slug": "rachana-sagar-physics-with-xtra-innings-class-12",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0685628001461390353.jpg",
        "title": "Physics with Xtra innings class 12",
        "u_by": 1,
        "u_date": 1573815859,
        "uploaded_by": "",
        "weight": "0.58",
        "width": null,
        "highlights": [
          {
            "field": "title",
            "snippet": "Physics with Xtra innings class 12"
          }
        ],
        "text_match": "33488996"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-23",
        "added_to_search": "Y",
        "author": "R K Gaur",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 1059,
        "book_type": 0,
        "category": 196,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461389164,
        "id": "931",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9788185018642",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": null,
        "price": 225,
        "publication": "DHMhanpan Rai",
        "publication_date": "2000",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 90,
        "sku": "A3AF848944",
        "slug": "engineering-physics",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0386530001461390353.jpg",
        "title": "Engineering physics",
        "u_by": 1,
        "u_date": 1569649830,
        "uploaded_by": "",
        "weight": "1.53",
        "width": null,
        "highlights": [
          {
            "field": "title",
            "snippet": "Engineering"
          }
        ],
        "text_match": "33488996"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-23",
        "added_to_search": "Y",
        "author": "Sisir Ranjan Majumder",
        "binding": "Hardback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 1045,
        "book_type": 0,
        "category": 276,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461329875,
        "id": "917",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9788185018642",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": null,
        "price": 160,
        "publication": "New Light",
        "publication_date": "2000",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 60,
        "sku": "DA20662829",
        "slug": "iit-solved-papers-mathematics-chemistry-physics-1988-to-2000",
        "subject": null,
        "thickness": null,
        "thumb": "medium/09901079001461389160.jpg",
        "title": "IIT Solved papers Mathematics chemistry physics 1988 to 2000",
        "u_by": 1,
        "u_date": 1461389164,
        "uploaded_by": "",
        "weight": "0.75",
        "width": null,
        "highlights": [
          {
            "field": "title",
            "snippet": "IIT Solved papers Mathematics chemistry physics"
          }
        ],
        "text_match": "33488996"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-22",
        "added_to_search": "Y",
        "author": "Partha Pratim Pal",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 1044,
        "book_type": 0,
        "category": 274,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461329866,
        "id": "916",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9789350370520",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": null,
        "price": 82,
        "publication": "Prashikshan",
        "publication_date": "2010",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 50,
        "sku": "ICA51ACC7C",
        "slug": "fundamentals-of-physicak-sciences-for-class-9-and-10",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0378544001461329871.jpg",
        "title": "Fundamentals Of Physicak Sciences for class 9 and 10",
        "u_by": 1,
        "u_date": 1569649830,
        "uploaded_by": "",
        "weight": "0.36",
        "width": null,
        "highlights": [
          {
            "field": "title",
            "snippet": "Fundamentals Of Physicak Sciences for class 9 and 10"
          }
        ],
        "text_match": "33488996"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-22",
        "added_to_search": "Y",
        "author": "Devi Prasad",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 917,
        "book_type": 0,
        "category": 309,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461070197,
        "id": "789",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "0",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9788173156243",
        "keywords": null,
        "landing_price": 0,
        "language": "Hindi",
        "new_book_qty": 0,
        "no_of_pages": "222",
        "price": 95,
        "publication": "Pravat Prakasan",
        "publication_date": "2012",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 60,
        "sku": "5D9C2608C5",
        "slug": "5-point-someone",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0896687001461326021.jpg",
        "title": "5 Point Someone",
        "u_by": 1,
        "u_date": 1461326026,
        "uploaded_by": "",
        "weight": "0.20",
        "width": null,
        "highlights": [
          {
            "field": "isbn",
            "snippet": "<mark>9788173156243</mark>"
          }
        ],
        "text_match": "33488993"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-19",
        "added_to_search": "Y",
        "author": "Devi Prasad",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 917,
        "book_type": 0,
        "category": 309,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461070197,
        "id": "789",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "0",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9788173156243",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": "348",
        "price": 290,
        "publication": "Narosa",
        "publication_date": "2006",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 60,
        "sku": "72F0A731EA",
        "slug": "introduction-to-numerical-analysis-2nd-edition",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0615097001461070192.jpg",
        "title": "Introduction To NUMERICAL ANALYSIS 2nd edition",
        "u_by": 1,
        "u_date": 1477399103,
        "uploaded_by": "",
        "weight": "0.50",
        "width": null,
        "highlights": [
          {
            "field": "isbn",
            "snippet": "<mark>9788173156243</mark>"
          }
        ],
        "text_match": "33357921"
      }
    }
  ]
}
```

## Fig 47 – Title Search, Search API

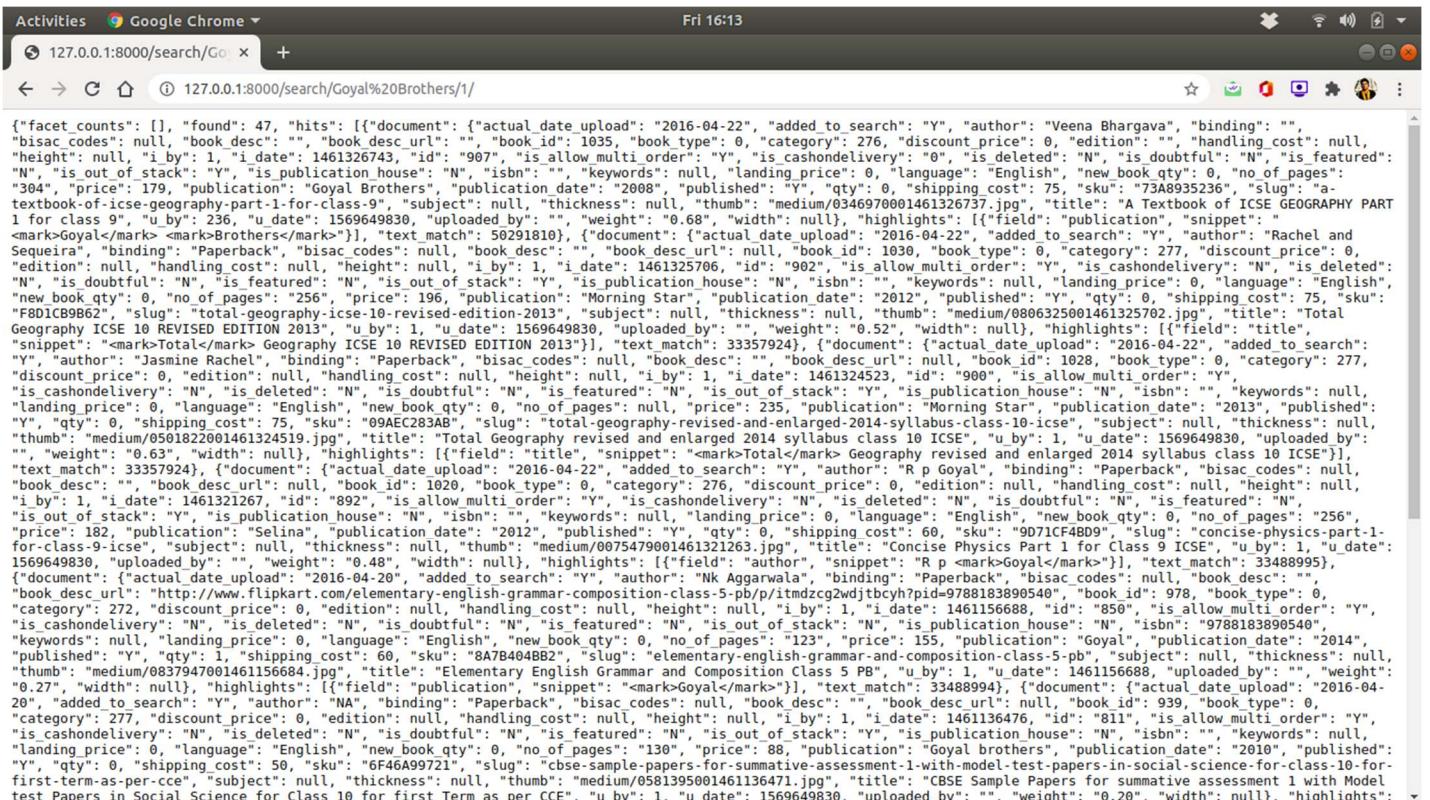
← → C ⌂ ① 127.0.0.1:8000/search/9788173156243/1

```
{
  "facet_counts": [],
  "found": 2,
  "hits": [
    {
      "document": {
        "actual_date_upload": "2016-04-22",
        "added_to_search": "Y",
        "author": "Chetan Bhagat",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": "http://www.flipkart.com/5-point-someone/pitmehmsnhxpucgm?pid=9788173156243",
        "book_id": 1032,
        "book_type": 0,
        "category": 224,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461326026,
        "id": "904",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "N",
        "is_deleted": "N",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9788173156243",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": "222",
        "price": 95,
        "publication": "Pravat Prakasan",
        "publication_date": "2012",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 60,
        "sku": "5D9C2608C5",
        "slug": "5-point-someone",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0896687001461326021.jpg",
        "title": "5 Point Someone",
        "u_by": 1,
        "u_date": 1461326026,
        "uploaded_by": "",
        "weight": "0.20",
        "width": null,
        "highlights": [
          {
            "field": "isbn",
            "snippet": "<mark>9788173156243</mark>"
          }
        ],
        "text_match": "33488993"
      }
    },
    {
      "document": {
        "actual_date_upload": "2016-04-19",
        "added_to_search": "Y",
        "author": "Devi Prasad",
        "binding": "Paperback",
        "bisac_codes": null,
        "book_desc": "",
        "book_desc_url": null,
        "book_id": 917,
        "book_type": 0,
        "category": 309,
        "discount_price": 0,
        "edition": null,
        "handling_cost": null,
        "height": null,
        "i_by": 1,
        "i_date": 1461070197,
        "id": "789",
        "is_allow_multi_order": "Y",
        "is_cashondelivery": "0",
        "is_doubtful": "N",
        "is_featured": "N",
        "is_out_of_stack": "Y",
        "is_publication_house": "N",
        "isbn": "9788173156243",
        "keywords": null,
        "landing_price": 0,
        "language": "English",
        "new_book_qty": 0,
        "no_of_pages": "348",
        "price": 290,
        "publication": "Narosa",
        "publication_date": "2006",
        "published": "Y",
        "qty": 0,
        "shipping_cost": 60,
        "sku": "72F0A731EA",
        "slug": "introduction-to-numerical-analysis-2nd-edition",
        "subject": null,
        "thickness": null,
        "thumb": "medium/0615097001461070192.jpg",
        "title": "Introduction To NUMERICAL ANALYSIS 2nd edition",
        "u_by": 1,
        "u_date": 1477399103,
        "uploaded_by": "",
        "weight": "0.50",
        "width": null,
        "highlights": [
          {
            "field": "isbn",
            "snippet": "<mark>9788173156243</mark>"
          }
        ],
        "text_match": "33357921"
      }
    }
  ]
}
```

## Fig 48 – ISBN Search, Search API



#### Fig 49 – Author Search, Search API



**Fig 50 – Publisher Search, Search API**

**Figure 47 to 50**, shows **Search API**. This API was built using **TypeSence Server**. Basically, using this API, one can search for a book by its name, or author name, or publisher name, or isbn code. Our database table of books has been used for creating the schema and training the algorithm.

The API returns the books which are sorted by book id's in descending order. It is a **GET Method API**.

The search is fast & flexible, therefore making it easier for user to search for a book.

```

1 {"status":200,"output":[{"request_details":{"delivery_type":"FORWARD","drop_pincode":560036,"order_type":"PREPAID","invoice_value":197.0,"reference_number":37398,"pickup_pincode":700136,"item":"books","preference_array":[{"cp_name":"Aramex","combination_sla_rto":-96.82284886694403,"rto_rate":0,"priority":1,"price":-96.82284886694403,"account_id":135,"base_price":42.01,"account_code":"PREPAID","sla_breach":0,"cp_id":2},{"cp_name":"Delhivery","combination_sla_rto":-96.43033721779953,"rto_rate":0,"priority":2,"price":-96.43033721779953,"account_id":388,"base_price":47.2,"account_code":"Olvr_Surfc_500gm","sla_breach":0,"cp_id":4}, {"cp_name":"Delhivery","combination_sla_rto":-95.71640466135943,"rto_rate":0,"priority":3,"price":-95.71640466135943,"account_id":134,"base_price":56.64,"account_code":1,"sla_breach":0,"cp_id":4}, {"cp_name":"Delhivery","combination_sla_rto":-94.97978357028673,"rto_rate":0,"priority":4,"price":-94.97978357028673,"account_id":387,"base_price":66.38,"account_code":"Olvr_Air_500gm","sla_breach":0,"cp_id":4}, {"cp_name":"Fedex","combination_sla_rto":-94.6455058266993,"rto_rate":0,"priority":5,"price":-94.6455058266993,"account_id":132,"base_price":70.8,"account_code":"outbound","sla_breach":0,"cp_id":1}, {"cp_name":"SpotOn","combination_sla_rto":-77.6896076112471,"rto_rate":0,"priority":6,"price":-77.6896076112471,"account_id":247,"base_price":295.0,"account_code":outbound,"sla_breach":0,"cp_id":39}, {"cp_name":"Fedex","combination_sla_rto":278.142243877168,"rto_rate":0,"priority":7,"price":278.142243877168,"account_id":442,"base_price":5000,"account_code":Bulk_Outbound_Fedex,"sla_breach":0,"cp_id":1}, {"cp_name":"Fedex","combination_sla_rto":278.142243877168,"rto_rate":0,"priority":8,"price":278.142243877168,"account_id":443,"base_price":5000,"account_code":Bulk_Inbound_Fedex,"sla_breach":0,"cp_id":1}], "pincode_serviceable":true, "form_tax_details": [{"entry_tax":false, "form_name": ""}], "meta": [{"message": "SUCCESS", "status": 200, "success": true}], "reason": "OK"}]

```

**Fig 51 – ClickPost Recommendation API**

**Figure 51**, shows **ClickPost Recommendation API**. Basically, the API is build on top of ClickPost API. The API fetches the delivery cost of the item at a particular pincode. The lowest delivery cost of all is shown to the user along with the deliver time. Also, if a user wants faster delivery service then that is also displayed to the user. It is a **POST Method API**.

POST http://127.0.0.1:8000/api/v1/post/donation\_form

Body (2)

```

1 { "data" :
2   {
3     "mobile" : 8334827085,
4     "address" : "test_address_0103",
5     "landmark": "test_0103",
6     "country" : "test_0103",
7     "state"   : "test0103",
8     "city"    : "test0103",
9     "zipcode" : 1234560103,
10    "no_of_book": 2,
11    "no_of_cartons": 2,
12    "app_books_weight": 2.54321,
13    "category" : "test_0103",
14    "source"   : "test_0103",
15    "donor_name": "aayush_test_0103",
16    "pickup_date": "01-03-2019"
17  }

```

Status: 201 Created Time: 1192 ms Size: 302 B

```

1 {
2   "status": 200,
3   "message": "Donor Details of User Id- 214 , saved successfully."
4 }

```

**Fig 52 – Donation Form API**

**Figure 52,** shows the **Donation Form API.** Basically, this **application is used to save the details of the filled donation form to the database.**

Once a user fills up the donation form, where they have requested to donate their books then the **details filled by them are saved in the database using this API.**

It is a **POST Method API.**

The API works only when the **user is logged-in.** Therefore, **token has to be passed in the Headers.**

```

POST http://127.0.0.1:8000/api/v1/post/get_razorpayid
Body JSON (application/json)
1  {
2    "data" :
3    {
4      "ref_id" : 515776,
5      "amount" : 100
6    }
7  }
8

```

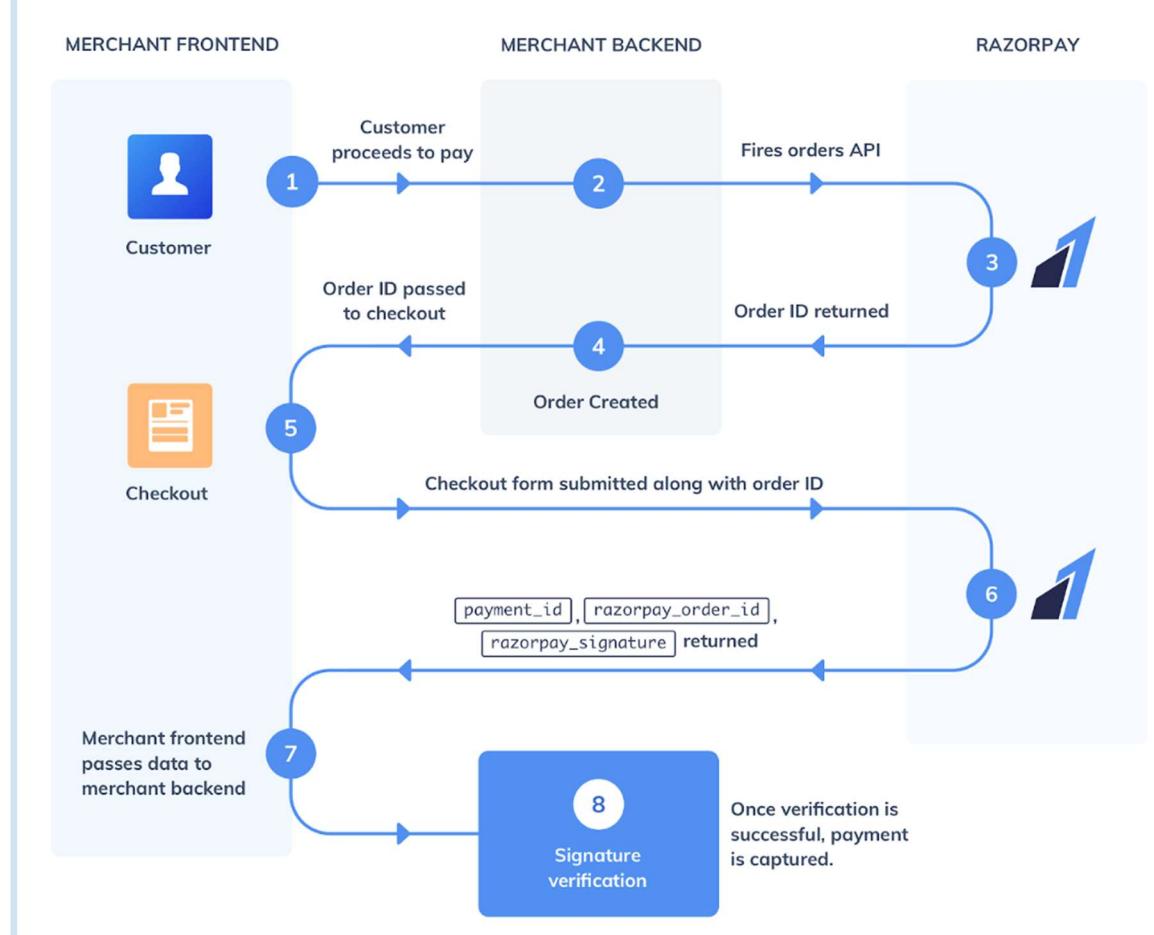
Body Cookies (1) Headers (7) Test Results Status: 201 Created Time: 913 ms Size: 300 B Download

```

1  {
2    "status": 200,
3    "message": "success",
4    "output": {
5      "rp_id": "order_C1FAeIgX3kY0j5"
6    }
7  }

```

**Fig 53 – Generate RazorPay ID API**

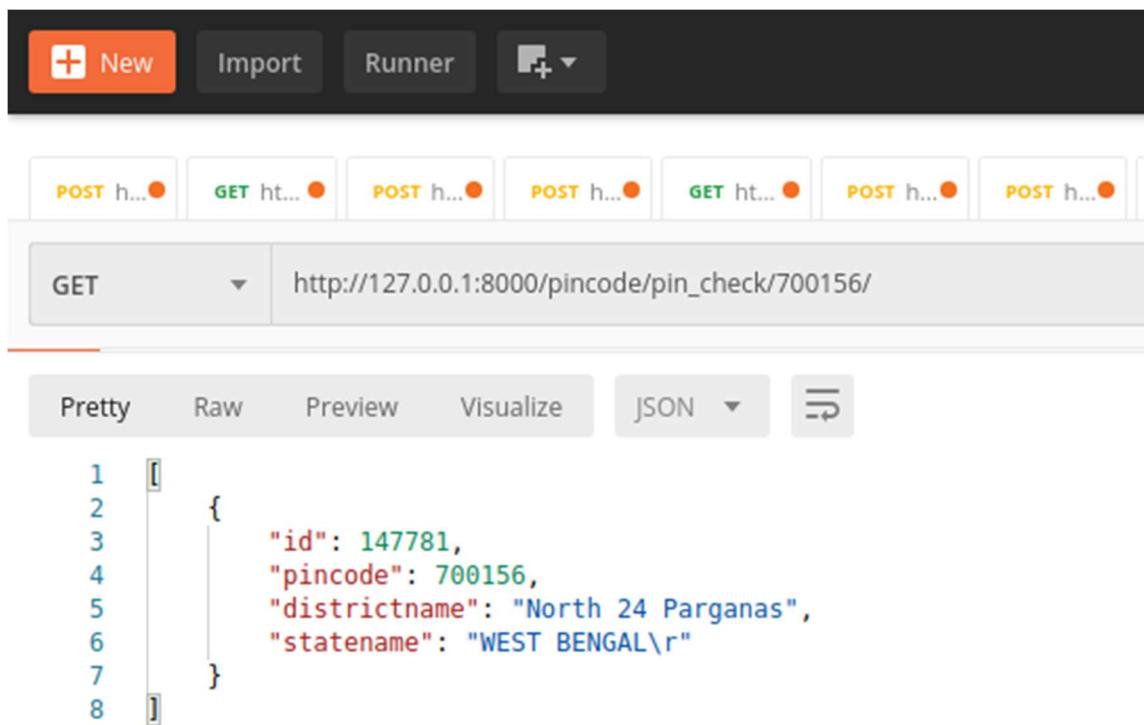


**Fig 54 – RazorPay Payment Gateway Flowchart**

**Figure 53**, shows the **Generate RazorPay ID API**. Basically, **Razorpay is the Payment Gateway we have integrated in our website**. So, in order to use their payment gateway as clearly seen in their Flowchart on **Figure 54**, we have to generate a **RazorPay ID** and that has to be sent once the user **order\_id** has been generated & user has clicked to proceed to checkout.

This, **RazorPay ID** is sent to the **RazorPay**, using which after the payment is complete, **RazorPay** returns the **payment\_id**, **razorpay\_order\_id** & **razorpay\_signature**.

This is the utility of the **API**. It is a **POST Method API**.



The screenshot shows the Postman application interface. At the top, there are buttons for '+ New', 'Import', 'Runner', and a settings icon. Below the header, there are several small thumbnail previews of other requests. The main request area shows a 'GET' method selected from a dropdown, and the URL 'http://127.0.0.1:8000/pincode/pin\_check/700156/' entered into the input field. Below the URL, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', 'JSON', and a copy icon. The JSON response is displayed as follows:

```
1 [  
2 {  
3   "id": 147781,  
4   "pincode": 700156,  
5   "districtname": "North 24 Parganas",  
6   "statename": "WEST BENGAL\r"  
7 }]  
8 ]
```

Fig 55 – PinCheck API - 1

The screenshot shows the Postman application interface. At the top, there are buttons for 'New' (orange), 'Import', 'Runner', and a dropdown menu. Below the header, there are several cards representing different API requests, mostly POST methods. In the main area, a 'GET' request is selected, and the URL is set to `http://127.0.0.1:8000/pincode/pin_check/741235`. The response is displayed in a JSONpretty format:

```
1 [  
2 {  
3     "id": 147208,  
4     "pincode": 741235,  
5     "districtname": "Nadia",  
6     "statename": "WEST BENGAL\r"  
7 }  
8 ]
```

**Fig 56 – PinCheck API - 2**

**Figure 55 & 56,** shows the PinCheck API. Basically, using this API, the details of a pincode are returned like the pincode\_id, pincode, districtname & statename.

These API is used when a user is adding address to its profile. Using this API, automatically the details of the Pincode are returned.

**It is a GET Method API.**

**NOTE for Backend API's-**

*For all the API's, the report consists of their URL's, passed Input Data, Headers, Functionality & the Output. The codes of the API couldn't be added to the report due to a large number of files for each Application. Although, the codes though have been presented during the Project Demonstration on 10<sup>th</sup> August' 2020.*

# CHATBOT

```
Activities Terminal ▾ Wed 02:55
aayush_linux@aayush-linux: ~/Downloads/college-project/cproject-bot_code

File Edit View Search Terminal Help
- donation_queries_form
- general_queries_form
(base) aayush_linux@aayush-linux:~/Downloads/college-project/cproject-bot_code$ rasa train
Training Core model...
/home/aayush_linux/anaconda3/lib/python3.7/site-packages/rasa/core/policies/ensemble.py:310: FutureWarning: 'KerasPolicy' is deprecated and will be removed in version 2.0. Use 'TEDPolicy' instead.
  policy_object = constr_func(**policy)
/home/aayush_linux/anaconda3/lib/python3.7/site-packages/rasa/utils/common.py:351: UserWarning: Found policies ['KerasPolicy', 'TEDPolicy'] with same priority 1 in PolicyEnsemble. When personalizing priorities, be sure to give all policies different priorities.
  More info at https://rasa.com/docs/rasa/core/policies/
Processed Story Blocks: 100% | 10/10 [00:00<00:00, 2382.18it/s, # trackers=1]
Processed Story Blocks: 100% | 10/10 [00:00<00:00, 366.51it/s, # trackers=10]
Processed Story Blocks: 100% | 10/10 [00:00<00:00, 72.01it/s, # trackers=50]
Processed Story Blocks: 100% | 10/10 [00:00<00:00, 73.93it/s, # trackers=44]
Processed Story Blocks: 100% | 10/10 [00:00<00:00, 1832.85it/s, # actions=30]
Processed trackers: 100% | 510/510 [00:00<00:00, 741.80it/s, # actions=809]
2020-07-08 02:55:14.366710: E tensorflow/stream_executor/cuda/cuda_driver.cc:351] failed call to cuInit: UNKNOWN ERROR (303)
Model: "sequential"

Layer (type)          Output Shape         Param #
=====
Masking (Masking)    (None, 5, 43)        0
lstm (LSTM)          (None, 32)           9728
dense (Dense)        (None, 24)            792
activation (Activation) (None, 24)        0
=====
Total params: 10,520
Trainable params: 10,520
Non-trainable params: 0

Train on 809 samples
Epoch 1/100
809/809 [=====] - 5s 7ms/sample - loss: 3.0067 - accuracy: 0.3572
Epoch 2/100
809/809 [=====] - 0s 171us/sample - loss: 2.4650 - accuracy: 0.4412
```

```
Activities Terminal ▾ Wed 02:56
aayush_linux@aayush-linux: ~/Downloads/college-project/cproject-bot_code

File Edit View Search Terminal Help
Epoch 83/100
809/809 [=====] - 0s 166us/sample - loss: 0.2224 - accuracy: 0.9209
Epoch 84/100
809/809 [=====] - 0s 168us/sample - loss: 0.2479 - accuracy: 0.9122
Epoch 85/100
809/809 [=====] - 0s 164us/sample - loss: 0.2322 - accuracy: 0.9147
Epoch 86/100
809/809 [=====] - 0s 165us/sample - loss: 0.2114 - accuracy: 0.9209
Epoch 87/100
809/809 [=====] - 0s 167us/sample - loss: 0.1731 - accuracy: 0.9370
Epoch 88/100
809/809 [=====] - 0s 165us/sample - loss: 0.2702 - accuracy: 0.8986
Epoch 89/100
809/809 [=====] - 0s 180us/sample - loss: 0.2571 - accuracy: 0.9110
Epoch 90/100
809/809 [=====] - 0s 167us/sample - loss: 0.2545 - accuracy: 0.9073
Epoch 91/100
809/809 [=====] - 0s 168us/sample - loss: 0.2419 - accuracy: 0.9098
Epoch 92/100
809/809 [=====] - 0s 167us/sample - loss: 0.2612 - accuracy: 0.9085
Epoch 93/100
809/809 [=====] - 0s 167us/sample - loss: 0.2484 - accuracy: 0.9110
Epoch 94/100
809/809 [=====] - 0s 171us/sample - loss: 0.2136 - accuracy: 0.9221
Epoch 95/100
809/809 [=====] - 0s 167us/sample - loss: 0.2536 - accuracy: 0.9122
Epoch 96/100
809/809 [=====] - 0s 167us/sample - loss: 0.2104 - accuracy: 0.9234
Epoch 97/100
809/809 [=====] - 0s 167us/sample - loss: 0.2083 - accuracy: 0.9271
Epoch 98/100
809/809 [=====] - 0s 165us/sample - loss: 0.2110 - accuracy: 0.9246
Epoch 99/100
809/809 [=====] - 0s 171us/sample - loss: 0.2201 - accuracy: 0.9197
Epoch 100/100
809/809 [=====] - 0s 167us/sample - loss: 0.1900 - accuracy: 0.9370
Processed trackers: 100% | 510/510 [00:00<00:00, 734.33it/s, # actions=809]
Epochs: 20% | 20/100 [00:17<00:55, 1.44it/s, t_loss=0.087, loss=0.011, acc=1.000]
```

Activities Terminal ▾ Wed 02:57  
aayush\_linux@aayush-linux: ~/Downloads/college-project/cproject-bot\_code

```

File Edit View Search Terminal Help
cpool 99/100
809/809 [=====] - 0s 167us/sample - loss: 0.2545 - accuracy: 0.9073
Epoch 91/100
809/809 [=====] - 0s 168us/sample - loss: 0.2419 - accuracy: 0.9098
Epoch 92/100
809/809 [=====] - 0s 167us/sample - loss: 0.2612 - accuracy: 0.9085
Epoch 93/100
809/809 [=====] - 0s 167us/sample - loss: 0.2484 - accuracy: 0.9110
Epoch 94/100
809/809 [=====] - 0s 171us/sample - loss: 0.2136 - accuracy: 0.9221
Epoch 95/100
809/809 [=====] - 0s 167us/sample - loss: 0.2536 - accuracy: 0.9122
Epoch 96/100
809/809 [=====] - 0s 167us/sample - loss: 0.2104 - accuracy: 0.9234
Epoch 97/100
809/809 [=====] - 0s 167us/sample - loss: 0.2083 - accuracy: 0.9271
Epoch 98/100
809/809 [=====] - 0s 165us/sample - loss: 0.2110 - accuracy: 0.9246
Epoch 99/100
809/809 [=====] - 0s 171us/sample - loss: 0.2201 - accuracy: 0.9197
Epoch 100/100
809/809 [=====] - 0s 167us/sample - loss: 0.1900 - accuracy: 0.9370
Processed trackers: 100% | 510/510 [00:00<00:00, 734.33it/s, # actions=809]
Epochs: 100% | 100/100 [00:55<00:00, 1.82it/s, t_loss=0.067, loss=0.002, acc=1.000]
2020-07-08 02:56:40 INFO rasa.utils.tensorflow.models - Finished training.
Processed trackers: 100% | 10/10 [00:00<00:00, 1729.75it/s, # actions=30]
2020-07-08 02:56:40 INFO rasa.core.agent - Persisted model to '/tmp/tmpsndoieo3/core'
Core model training completed.
Training NLU model...
2020-07-08 02:56:41 INFO rasa.nlu.training_data.training_data - Training data stats:
- intent examples: 101 (11 distinct intents)
- Found intents: 'donation', 'greet', 'out_of_scope', 'bot_challenge', 'insult', 'goodbye', 'payment', 'general', 'delivery', 'inform', 'reset'
- Number of response examples: 0 (0 distinct response)
- entity examples: 19 (4 distinct entities)
- found entities: 'General', 'Delivery', 'Donation', 'Payment'

/home/aayush_linux/anaconda3/lib/python3.7/site-packages/rasa/utils/common.py:351: UserWarning: Intent 'reset' has only 1 training examples! Minimum is 2, training may fail.
```

Activities Terminal ▾ Wed 02:57  
aayush\_linux@aayush-linux: ~/Downloads/college-project/cproject-bot\_code

```

File Edit View Search Terminal Help
Processed trackers: 100% | 10/10 [00:00<00:00, 1729.75it/s, # actions=30]
2020-07-08 02:56:40 INFO rasa.core.agent - Persisted model to '/tmp/tmpsndoieo3/core'
Core model training completed.
Training NLU model...
2020-07-08 02:56:41 INFO rasa.nlu.training_data.training_data - Training data stats:
- intent examples: 101 (11 distinct intents)
- Found intents: 'donation', 'greet', 'out_of_scope', 'bot_challenge', 'insult', 'goodbye', 'payment', 'general', 'delivery', 'inform', 'reset'
- Number of response examples: 0 (0 distinct response)
- entity examples: 19 (4 distinct entities)
- found entities: 'General', 'Delivery', 'Donation', 'Payment'

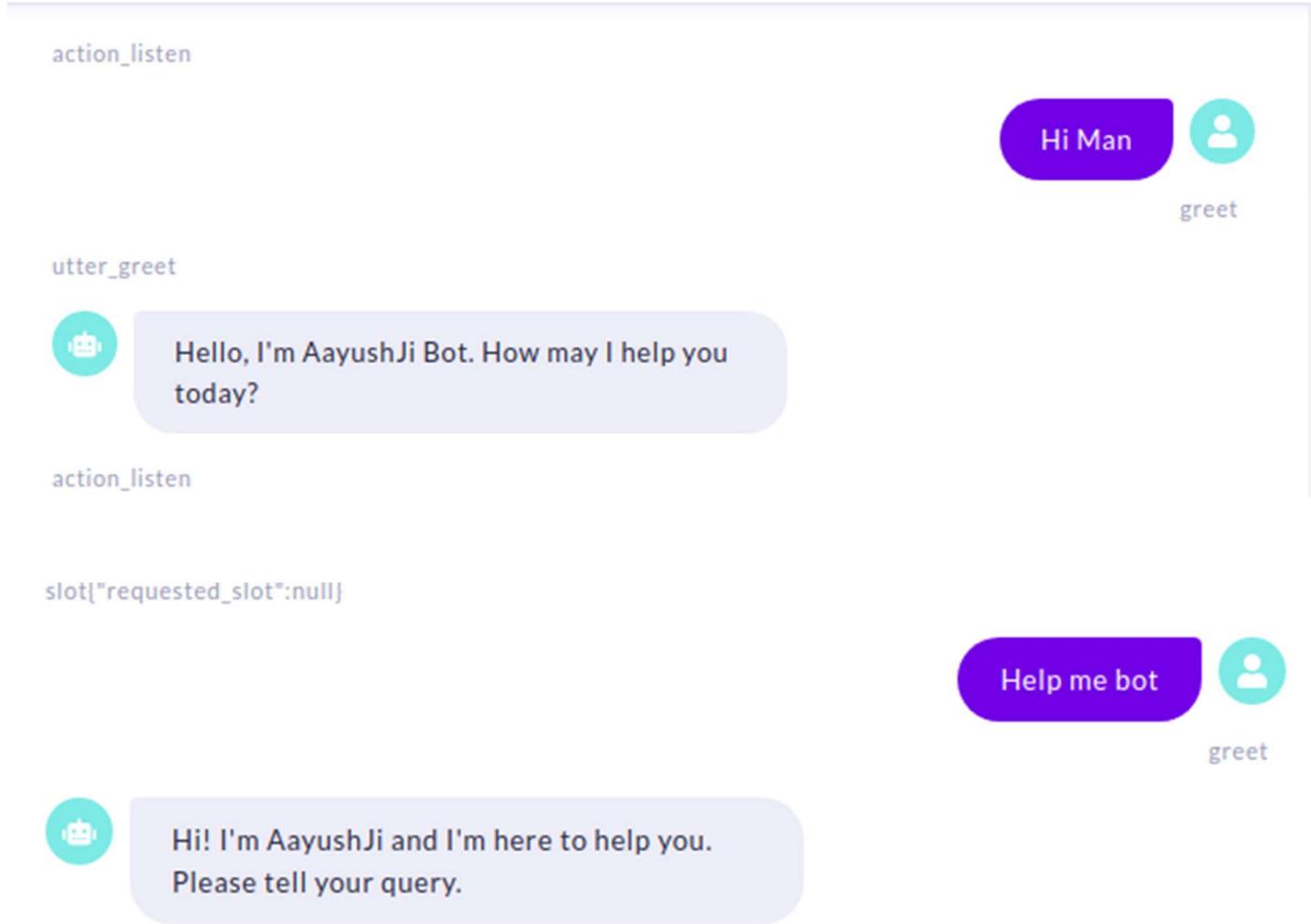
/home/aayush_linux/anaconda3/lib/python3.7/site-packages/rasa/utils/common.py:351: UserWarning: Intent 'reset' has only 1 training examples! Minimum is 2, training may fail.
2020-07-08 02:56:41 INFO rasa.nlu.model - Starting to train component WhitespaceTokenizer
2020-07-08 02:56:41 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:56:41 INFO rasa.nlu.model - Starting to train component RegexFeaturizer
2020-07-08 02:56:41 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:56:41 INFO rasa.nlu.model - Starting to train component LexicalSyntacticFeaturizer
2020-07-08 02:56:41 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:56:41 INFO rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2020-07-08 02:56:41 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:56:41 INFO rasa.nlu.model - Starting to train component CountVectorsFeaturizer
2020-07-08 02:56:41 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:56:41 INFO rasa.nlu.model - Starting to train component DIETClassifier
Epochs: 100% | 100/100 [00:38<00:00, 2.60it/s, t_loss=15.527, i_loss=7.082, e_loss=6.608, i_acc=0.861, e_f1=0.221]
2020-07-08 02:57:34 INFO rasa.utils.tensorflow.models - Finished training.
2020-07-08 02:57:34 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:57:34 INFO rasa.nlu.model - Starting to train component EntitySynonymMapper
2020-07-08 02:57:34 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:57:34 INFO rasa.nlu.model - Starting to train component ResponseSelector
2020-07-08 02:57:34 INFO rasa.nlu.selectors.response_selector - Retrieval intent parameter was left to its default value. This response selector will be trained on training examples combining all retrieval intents.
2020-07-08 02:57:34 INFO rasa.nlu.model - Finished training component.
2020-07-08 02:57:34 INFO rasa.nlu.model - Successfully saved model into '/tmp/tmpsndoieo3/nlu'
NLU model training completed.
Your Rasa model is trained and saved at '/home/aayush_linux/Downloads/college-project/cproject-bot_code/models/20200708-025734.tar.gz'.
(base) aayush_linux@aayush-linux:~/Downloads/college-project/cproject-bot_code$
```

**Fig 57 – Chatbot Being Trained**

**Figure 57**, covering Page- 44 & 45, show the **Chatbot Model being Trained**. The Chatbot has been build using the **RASA Framework**. The Model achieves an Accuracy of **93.70 % during the training**.

**Chatbot was built for the website so that it can answer user queries. General FAQ's can easily be answered by the Chatbot. Chatbot is also capable of answering to chitchat queries given by the user.**

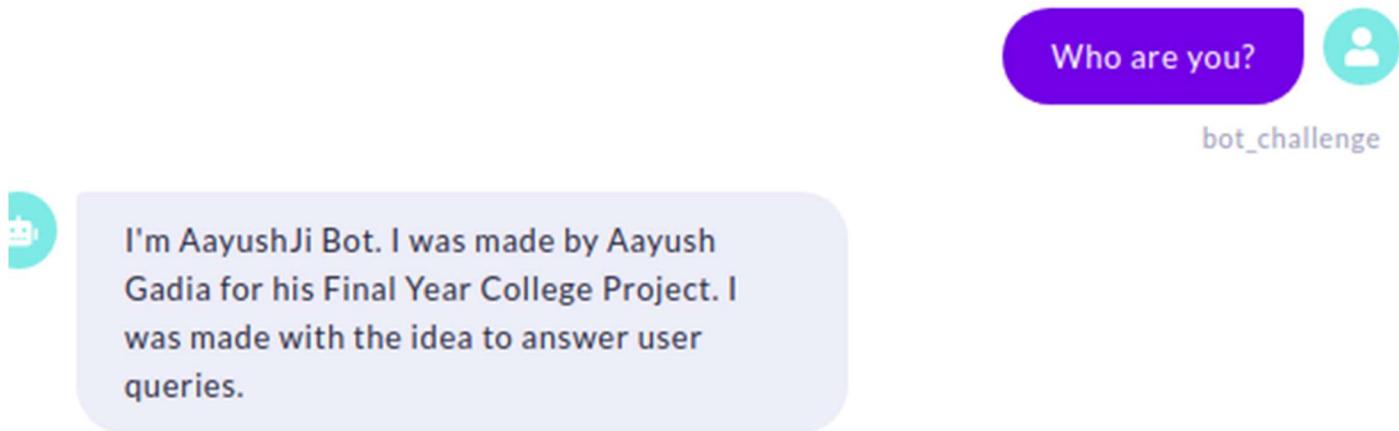
#### Talk to your bot (Interactive Learning)



**Fig 58 – Greeting Intent Reply**

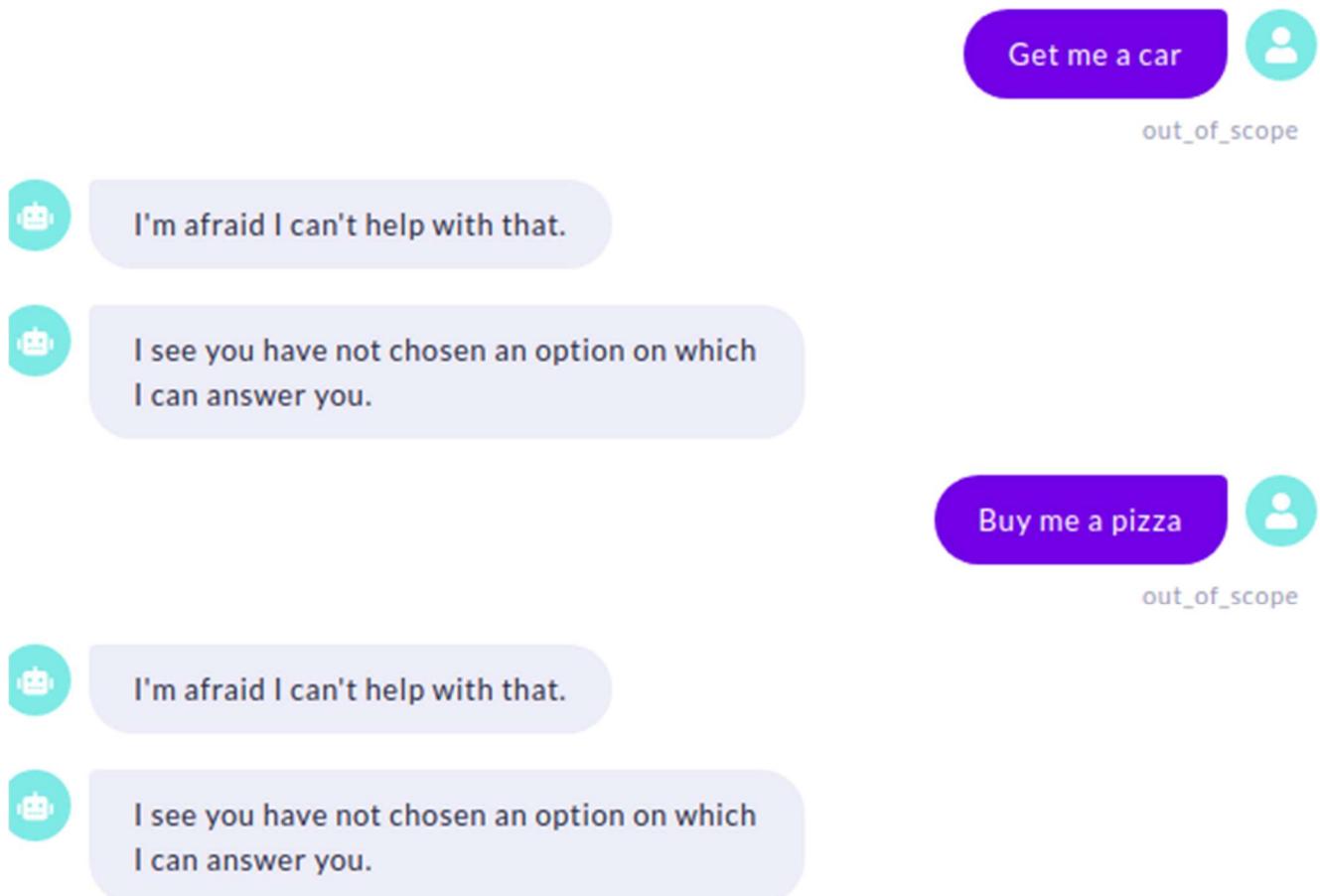
**Figure 58**, shows the **Greeting Intent Reply**, basically **Greeting Intent** is when the user is simply initiating the conversation with the bot by asking for help or simply greeting it. The bot has been trained on a custom dataset prepared by us for each intent.

The above figure shows how the bot would respond to the user input.



**Fig 59 – Bot Challenge Intent Reply**

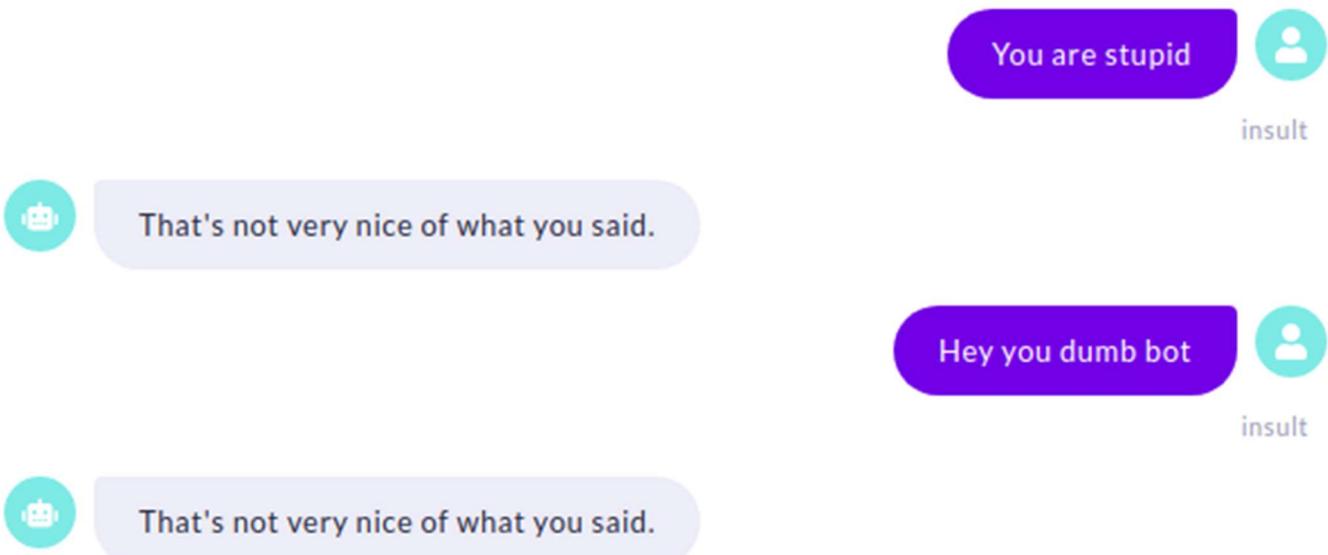
**Figure 59**, shows the **Bot Challenge Intent Reply**, basically **Bot Challenge Intent** is when the user is asking the bot about who the bot is or what it can do. The bot has been trained on a custom dataset prepared by us for each intent.



**Fig 60 – Out of Scope Intent Reply**

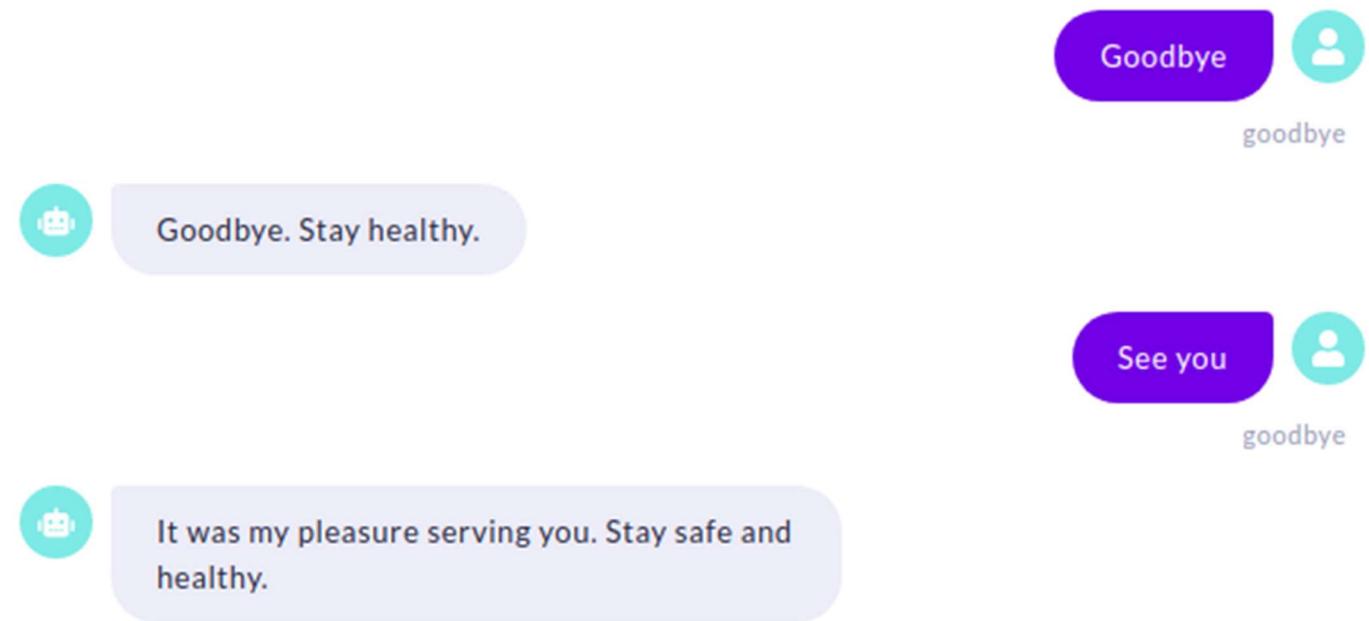
**Figure 60,** shows the **Out of Scope Intent Reply**, basically **Out of Scope Intent** is when the user is asking the bot to do things which the bot can never do or is outside the scope of the Bot abilities.

The bot has been trained on a custom dataset prepared by us for each intent.



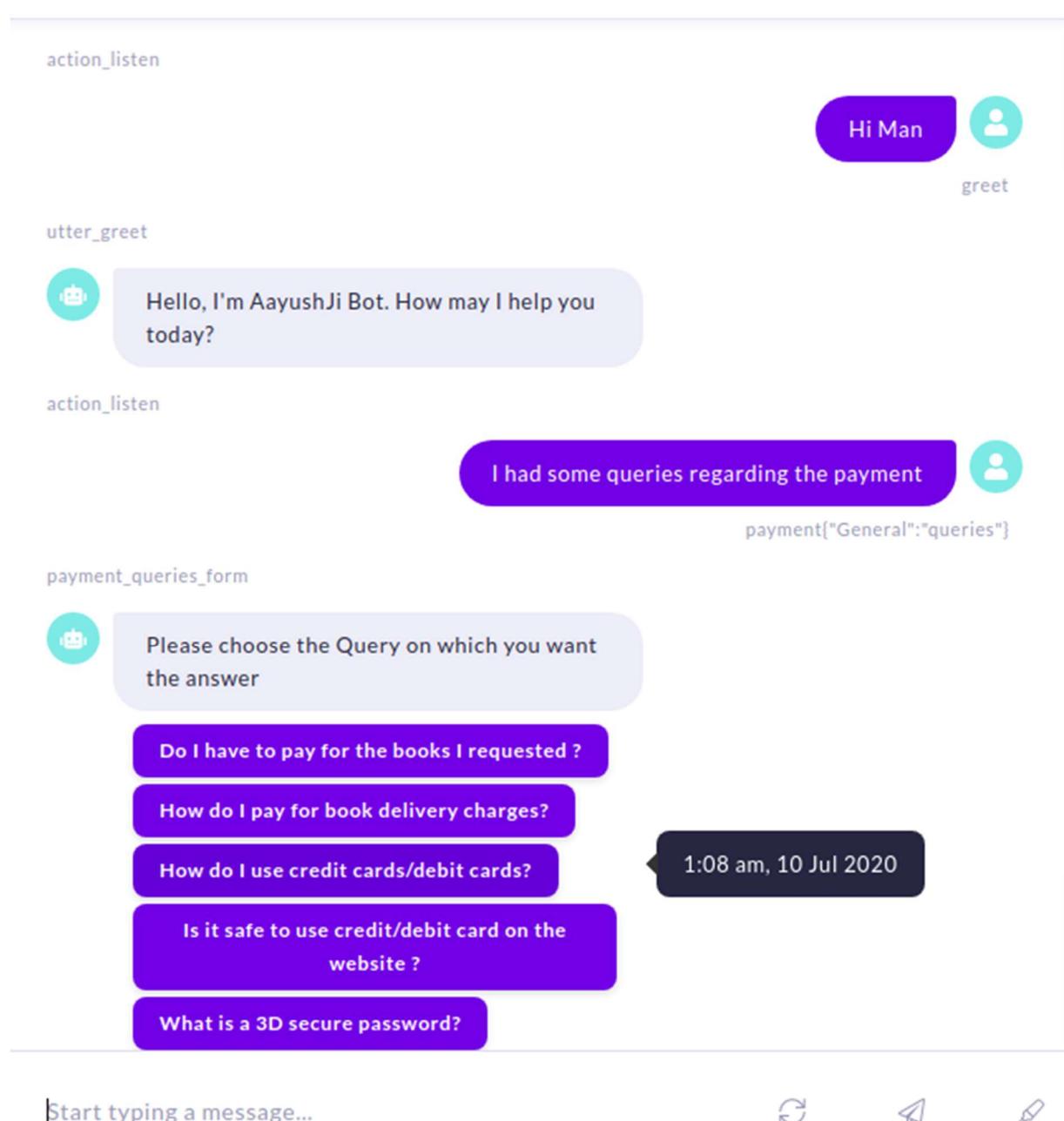
**Fig 61 – Insult Intent Reply**

**Figure 61,** shows the **Insult Intent Reply**, basically **Insult Intent** is when the user is **saying things to the bot which is harsh & abusive**. The **bot has been trained on a custom dataset prepared by us for each intent**.

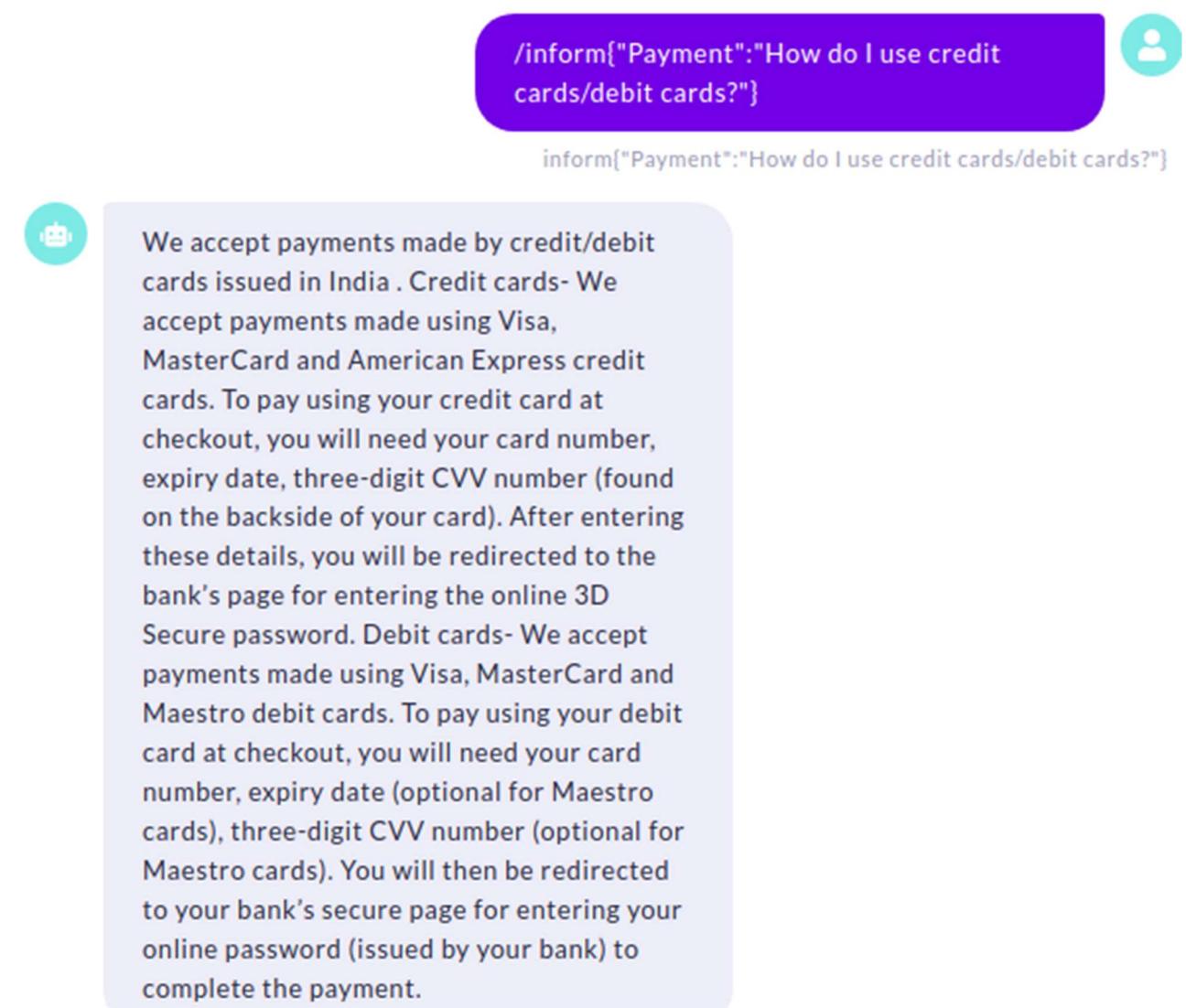


**Fig 62 – Goodbye Intent Reply**

**Figure 62**, shows the **Goodbye Intent Reply**, basically **Goodbye Intent** is when the user is **saying goodbye**. The **bot has been trained on a custom dataset prepared by us for each intent**.



**Fig 63 – Bot Conversation 1 : Payment Intent**



**Fig 64 – Bot Conversation 1 : Payment Intent**

**Figure 63 & 64,** shows the **Payment Intent Reply**, basically **Payment Intent** is when the user is **asking questions which are related to Payment** on the website. Once the intent is identified, the questions which the bot can answer is shown to the user, if any of the questions fulfills the user query then it is fine, otherwise the user can connect to the customer care. Here in the figure above the query had solved.

The bot has been trained on a custom dataset prepared by us for each intent.

## Talk to your bot (Interactive Learning)

action\_listen

Hi Man 

greet

utter\_greet

 Hello, I'm AayushJi Bot. How may I help you today?

action\_listen

I had some queries regarding the payment 

payment{"General":"queries"}

payment\_queries\_form

Please choose the Query on which you want the answer

Do I have to pay for the books I requested ?

How do I pay for book delivery charges?

How do I use credit cards/debit cards?

Is it safe to use credit/debit card on the website ?

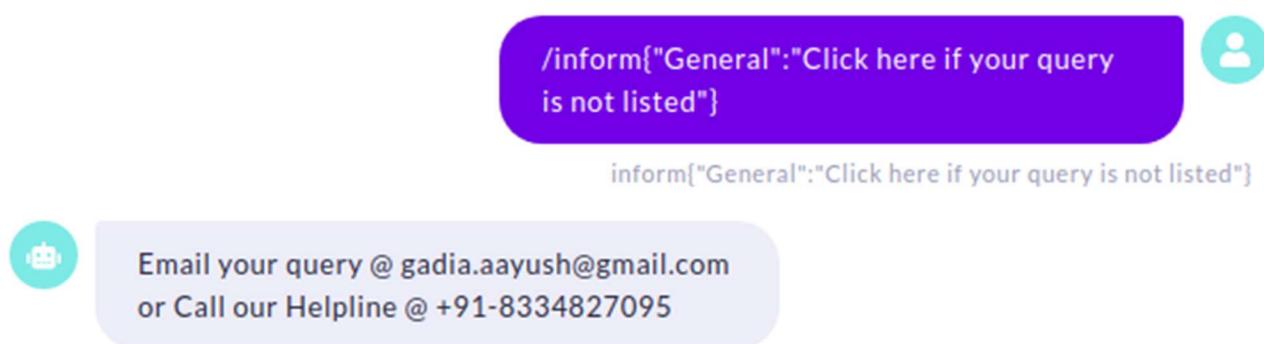
What is a 3D secure password?

1:08 am, 10 Jul 2020

Start typing a message...   

**Fig 65 – Bot Conversation 2 : Payment Intent**

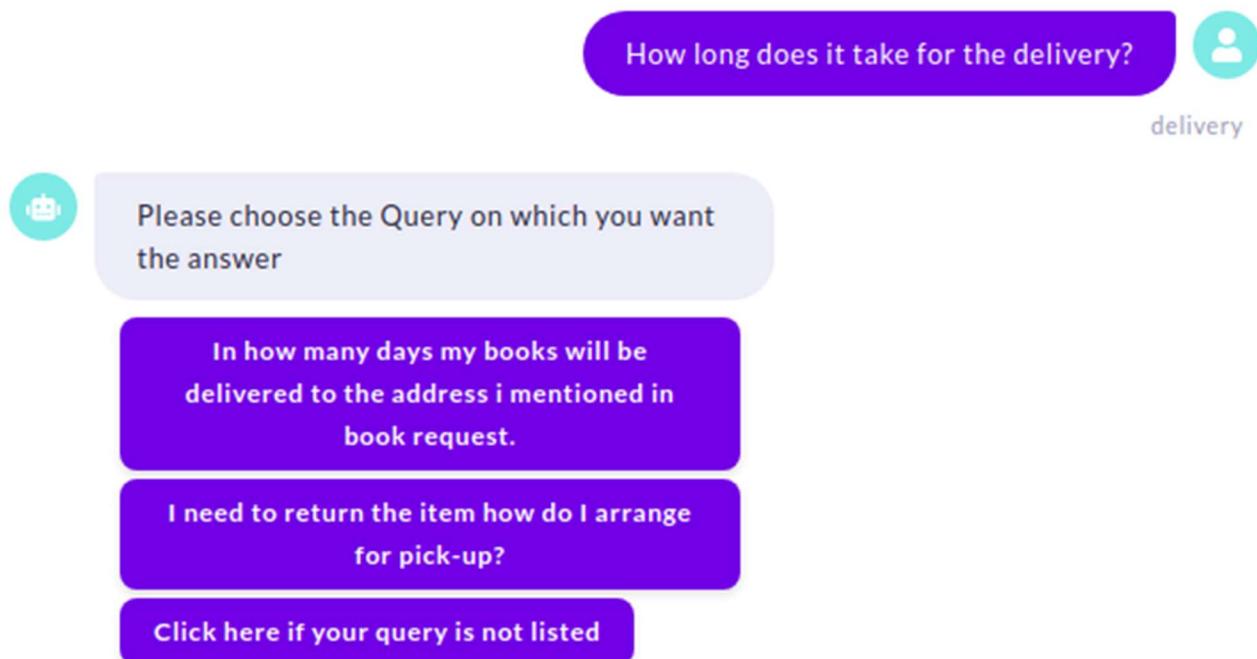
```
slot["requested_slot"]:"General"]
```



**Fig 66 – Bot Conversation 2 : Payment Intent**

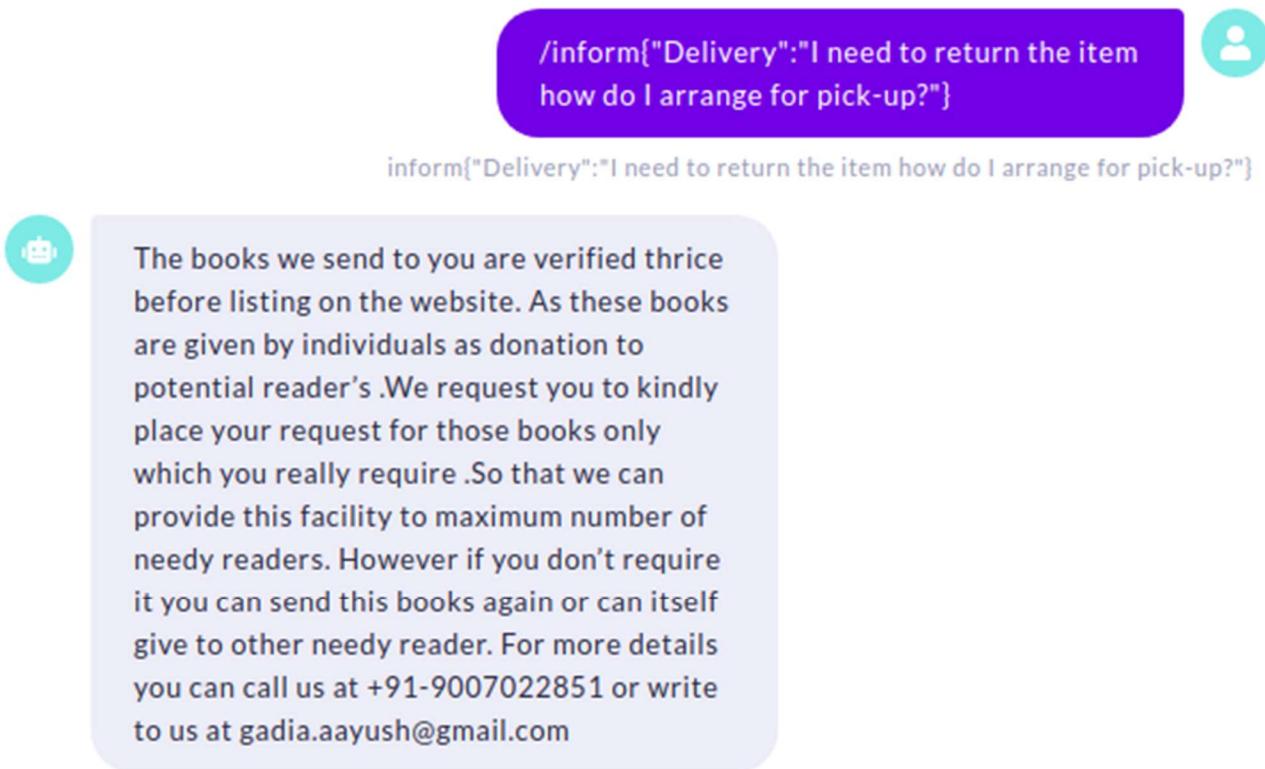
**Figure 65 & 66,** shows the **Payment Intent Reply**, basically **Payment Intent** is when the user is **asking questions which are related to Payment on the website**. Once the intent is identified, the questions which the bot can answer is shown to the user, if any of the questions fulfills the user query then it is fine, otherwise the user can connect to the customer care. Here in the figure above the query was not solved.

The **bot has been trained on a custom dataset prepared by us for each intent.**



```
slot["requested_slot"]:"Delivery"]
```

**Fig 67 – Bot Conversation : Delivery Intent**



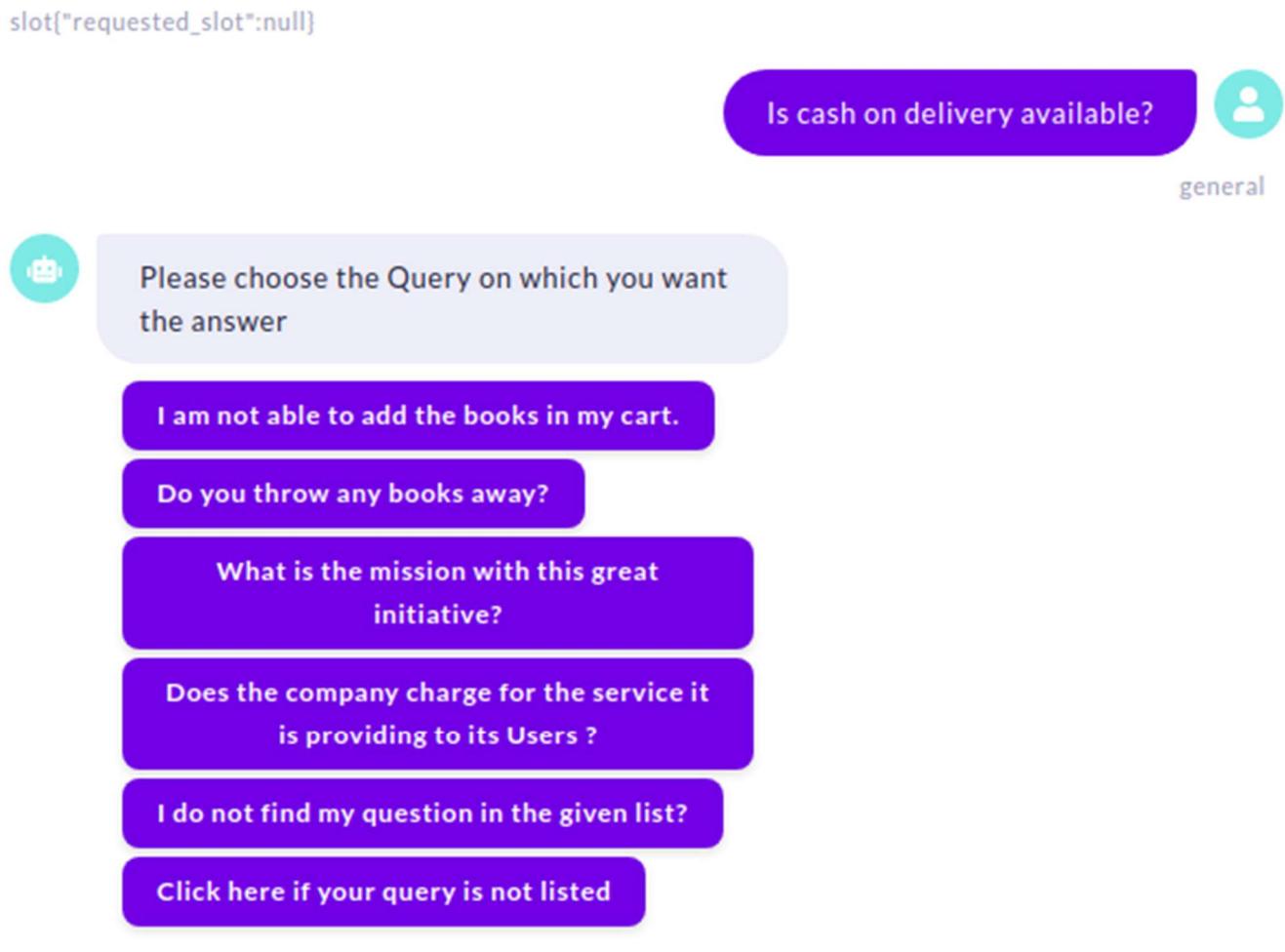
slot["Delivery":null]

slot["requested\_slot":null]

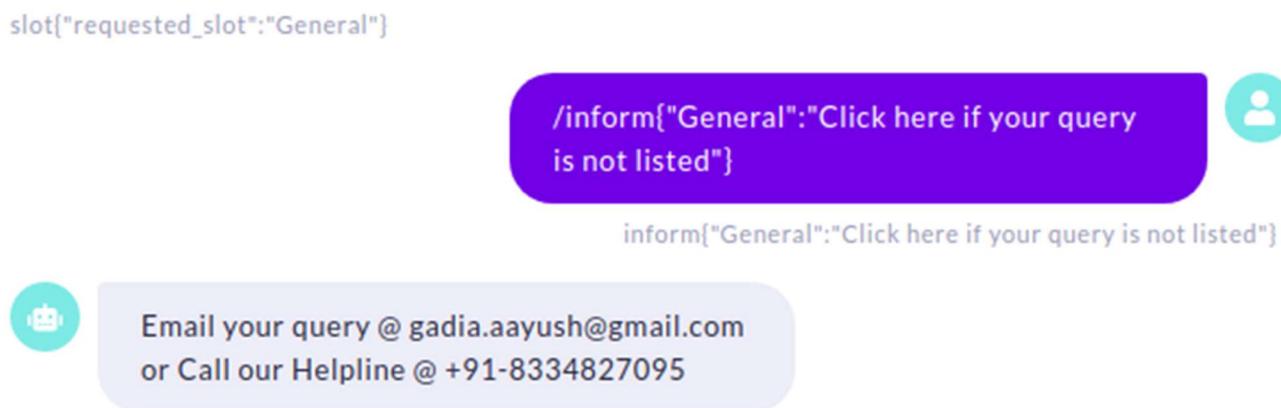
**Fig 68 – Bot Conversation : Delivery Intent**

**Figure 67 & 68,** shows the **Delivery Intent Reply**, basically **Delivery Intent** is when the user is **asking questions which are related to Delivery on the website**. Once the intent is identified, the questions which the bot can answer is shown to the user, if any of the questions fulfills the user query then it is fine, otherwise the user can connect to the customer care. Here in the figure above the query was solved.

**The bot has been trained on a custom dataset prepared by us for each intent.**



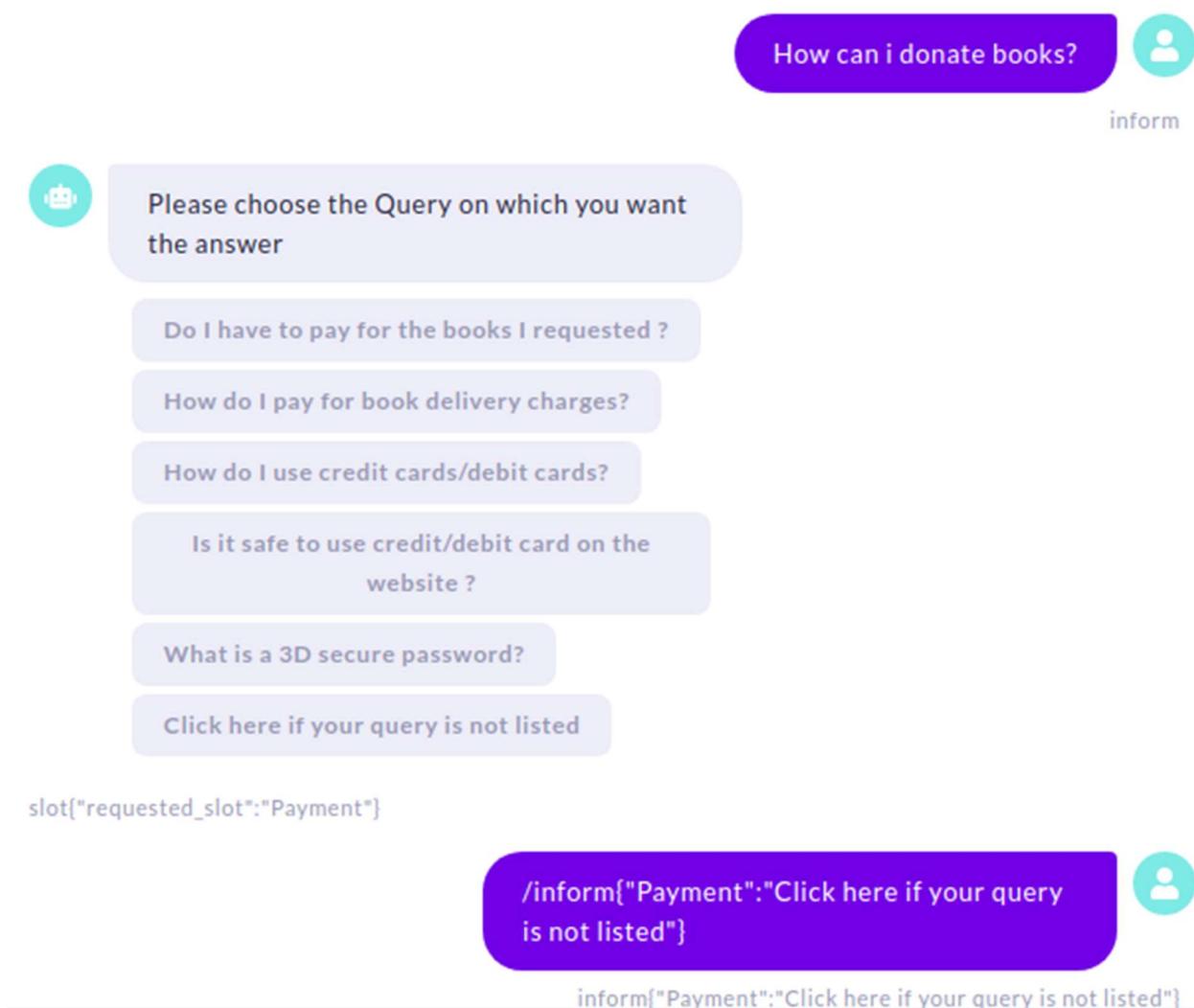
**Fig 69 – Bot Conversation : General Intent**



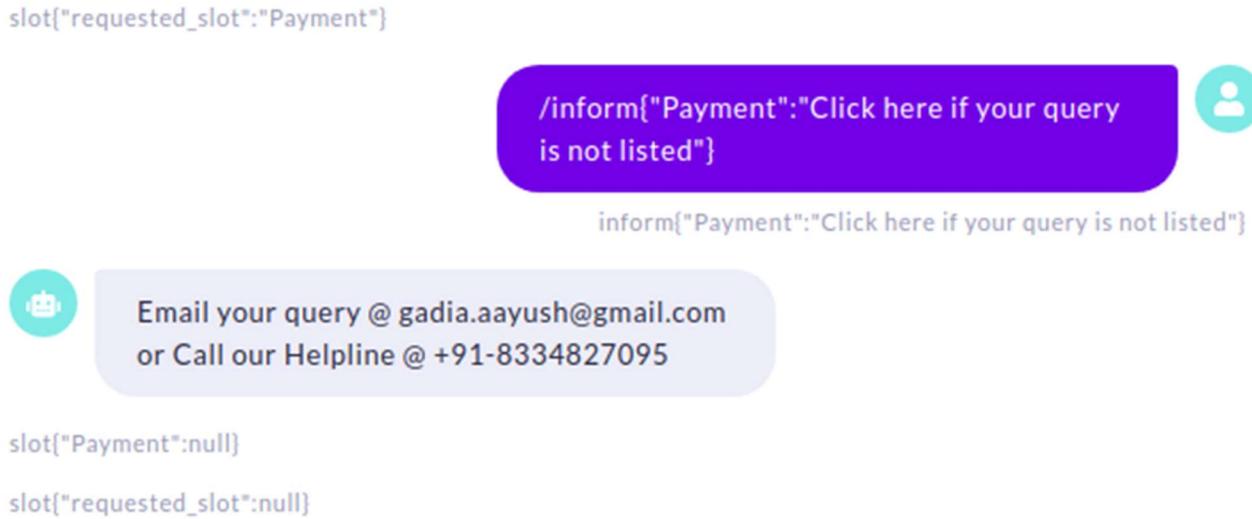
**Fig 70 – Bot Conversation : General Intent**

**Figure 69 & 70**, shows the **General Intent Reply**, basically General Intent is when the user is asking questions which are related to website or doubts regarding the website. Once the intent is identified, the questions which the bot can answer is shown to the user, if any of the questions fulfills the user query then it is fine, otherwise the user can connect to the customer care. Here in the figure above the query was not solved.

The bot has been trained on a custom dataset prepared by us for each intent.



**Fig 71 – Bot Conversation : Donate Intent**



**Fig 72 – Bot Conversation : Donate Intent**

**Figure 71 & 72,** shows that the user had typed **Donate Books** related intent, but the **bot identified it as a Payment related query intent and therefore, it classified it wrongly.**

**Therefore, this problem could be solved more, once more & more datasets are created and the chatbot is trained on it.**

**NOTE for Chatbot-**

*For the Chatbot, the report consists of the Dialogue Sessions of the Chatbot which shows how the Chatbot is replying to the different User Queries having different Intents. Also, the report consists of the screenshots showing the Chatbot being Trained and the Accuracy which was achieved. The codes of the Chatbot couldn't be added to the report due to a large number of files.*

*Although, the codes though have been presented during the Project Demonstration on 10<sup>th</sup> August' 2020.*

## Server Configuration

Following were the points covered during Server Configuration-

1. The entire FrontEnd Codes were made a bundle and was put on a different node on the server after configuring that node as per the configuration required for the FrontEnd.
2. The entire MySQL database was stored on a different node on the Server after creating a MYSQL node on the server.
3. The entire backend code along with the code of the chatbot was stored on a different node. That particular node was configured for executing the backend codes & the chatbot.
4. The URL's of the backend API's have been changed as per server's ip address for a given node.
5. Amazon S3 was used for storing the images of the books. Amazon Cloud Front was used as a Content Delivery Network (CDN) & it was also used for changing the URL's and hiding the actual URL's of the images stored in Amazon S3 bucket.
6. SSL certificates were enabled for the website by buying one, therefore the website was SSL certified and it was using HTTPS.
7. Also, the website was made PCI DSS Compliant.
8. Domain Names was registered.
9. A Testing Website was also created for debugging purpose.
10. All the codes on the website were kept in sync with GitHub, therefore once a developer adds a features or improves a feature or fixes a bug or anything, then just have to simply push the code on GitHub and then they can check that on Testing Website. Once, it is working fine, the code can be pushed to the Live Website GitHub Repository.

**11. Digital Ocean servers were used for hosting our website, be it frontend codes, databases, backend codes. Already, Digital Ocean was providing Hardware Firewall for the server on which our website was hosted.**

**Only, the software firewall had to be enabled. Therefore, Digital Ocean did have a software firewall option which was disabled by default, on enabling it, the software firewall also got enabled.**