

**Manav Rachna University, Faridabad**

**Jan-June( 2023)**

**“package tracking bot”**

**Submitted in partial fulfillment of the requirements**

**for the award of the degree of**

**Bachelor of Technology**

**in**

**Computer Science & Technology**

**Submitted by:-**

**Aryaman Rawat**

**Roll No. 2K20CSUN01014**

**Under the Supervision of**

**(Dr. Shalu)**

**Assistant Professor**

**Cloud Computing**

**Table of Contents**

1. Introduction to Project
2. User Requirements Specification
3. Use-case Diagram
4. Class Diagram
5. Data-Flow Diagram
6. E-R diagram
7. H/W and S/W Requirement
8. Working of the Bot
9. Application of Project
10. Input Data and Validation
11. Team Contribution
12. Project Vision
13. Conclusion
14. References

**SUPERVISER CERTIFICATE**

This is to certify that the Project submitted as “**Package tracer Bot**” submitted by Mr Aryaman Rawat to the Department of Computer Science & Engineering of Manav Rachna University of Science & Technology, Faridabad, Haryana

In my opinion, this work fulfills the requirement for which it has been submitted. This project report has not been submitted to any other university or institution for any degree.

**Dr. Shalu**

**(Supervisor)**

**CANDIDATE’S DECLARATION**

I hereby certify that the Project which is being presented in the project report entitled “**Package Tracer Bot**” is an authentic record of our own work carried out under the supervision of Dr. Shalu, Assisstant Professor. The matter presented in this project report has not been submitted by me for the award of any other degree in any other Institute / University.

**Akhil**

**(2K20CSUN01010)**

**Aryaman**

**(2K20CSUN01014)**

**Sachin**

**(2K20CSUN01089)**

**Acknowledgment**

We have taken a lot of effort into this project. However, completing this project would not have been possible without the support and guidance of a lot of individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to Dr. Shalu for her guidance and supervision. We would like to thank her for providing the necessary information and resources for this project. We would like to express our gratitude towards our parents & our friends for their kind co-operation and encouragement which help us a lot in completing this project. Our thanks and appreciations also go to our colleague in developing the project. Thank you to all the people who have willingly helped us out with their abilities.

**1. Introduction**

Package Tracer Bot is an innovative project that aims to make package tracking more convenient and accessible for everyone. With the rise of e-commerce and online shopping, more and more people are relying on package deliveries for their daily needs. However, tracking the packages can be a cumbersome task, especially if you have multiple packages being shipped from different vendors. That's where our bot comes in - it is a one-stop solution that can track all your packages from various carriers and provide you with real-time updates on their status.

One of the key benefits of using our Package Tracer Bot is its ease of use. You don't need to install any software or download any apps to use our bot. Simply connect with our bot on your favorite messaging platform, provide your tracking number, and our bot will take care of the rest. Our bot is available 24/7, and you can access it from anywhere, anytime. With our Package Tracer Bot, you can sit back, relax, and let our bot take care of your package tracking needs.

**2. User Requirement Specification**

For the package tracing bot on Telegram, the following functionalities must be provided to the users:

2.1 Users must be able to:

• Create an account.

• Manage their account.

• Log in to the system.

• Enter their package tracking number.

• Get real-time updates on their package status.

• View the estimated delivery time.

• Receive notifications on the status of their package.

• Contact customer support for any issues related to the package.

• Provide feedback on their experience with the bot.

2.2 The functions afforded by the package management system provide user with the ability to using a graphical interface:

• Update package details such as delivery address, receiver information, and package weight.

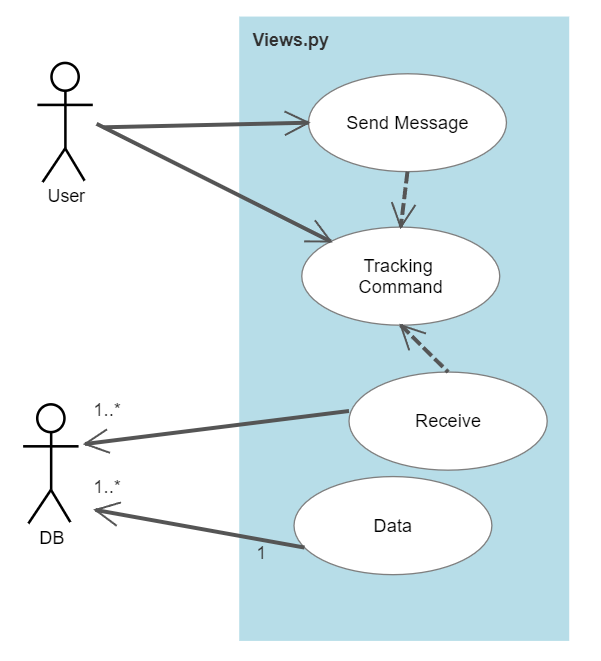
• View package history, including previous delivery attempts.

• Cancel a package delivery request.

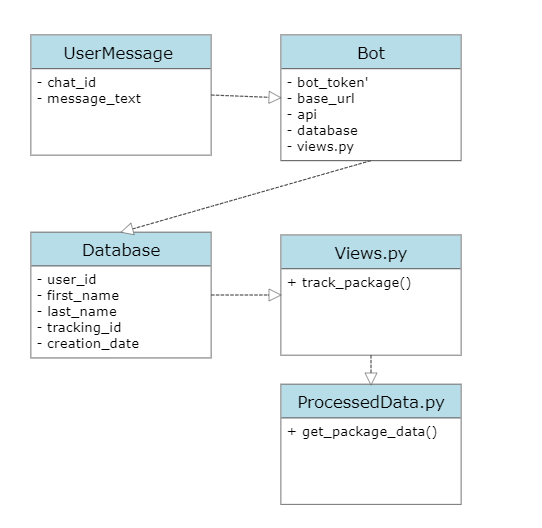
• Request for a package pickup.

• Track multiple packages at once.

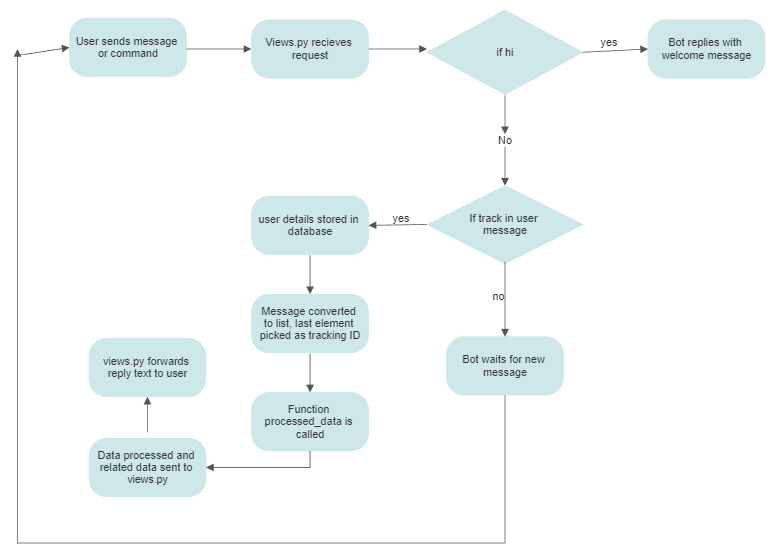
**3. USE CASE DIAGRAM**



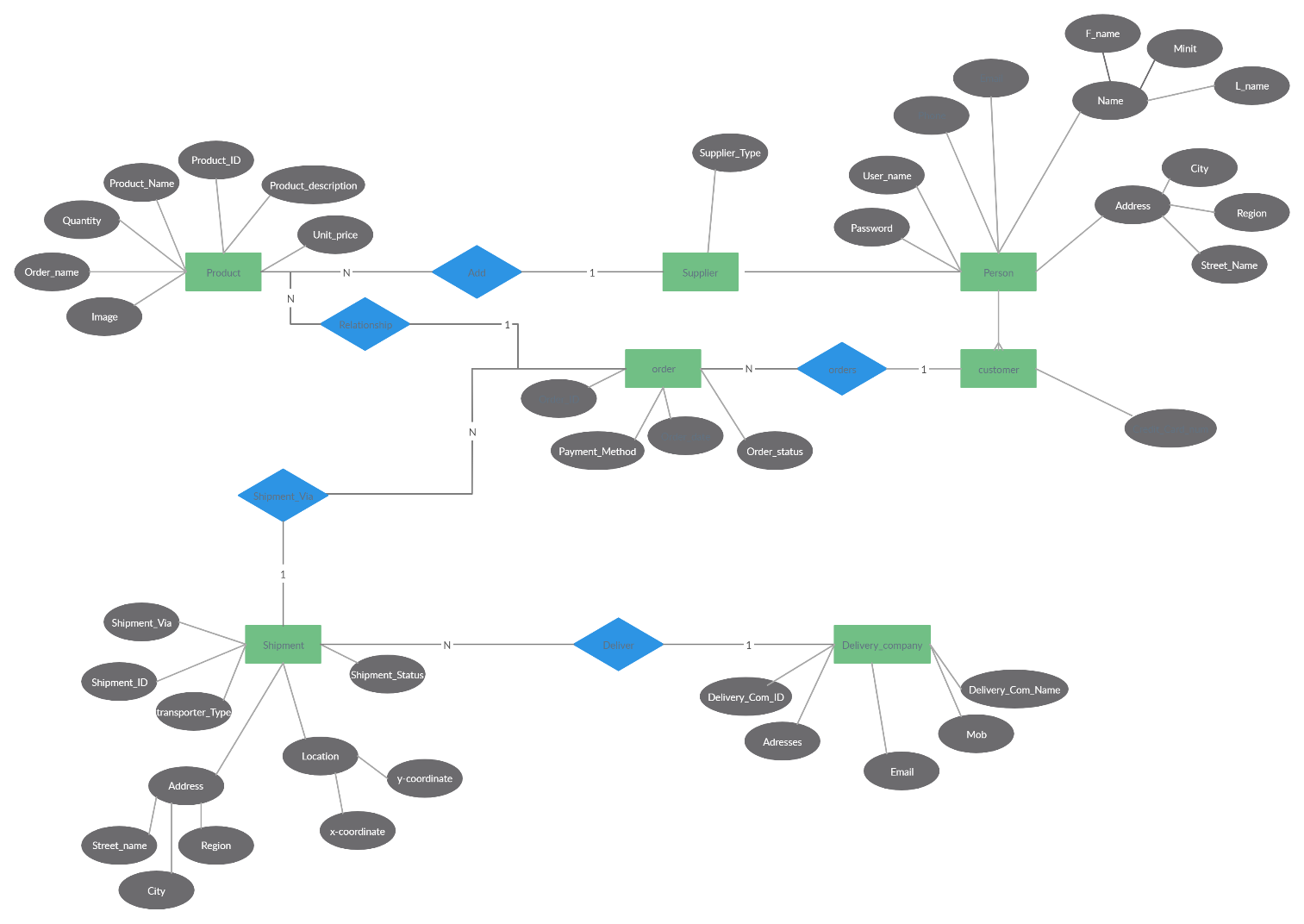
**4.Class diagram**



**5. Data flow diagram**



**6. E-R Diagram**



**7. S/W and H/W Requirements**

1. Access to a messaging platform: You need to have access to a messaging platform where the bot is available, such as Telegram or Facebook Messenger.
2. A tracking id: You need to have the tracking number of the package you want to track. This is usually provided by the courier or shipping company when you place an order.
3. Internet connectivity: You need to be connected to the internet to use the bot and track your package.
4. Ability to receive notifications: If the bot offers the feature, you may need to enable notifications to receive updates about your package.
5. Device compatibility: You need to make sure that the device you are using to access the messaging platform is compatible with the bot. Some bots may only work on certain devices or platforms.

Software Requirements:

1. Extensions: Telegram, Django, traceback, Json.

**8. Applications of Package tracking bot**

1. E-commerce: Package tracking bot can be used by e-commerce platforms to help customers track their orders. Customers can receive notifications about the status of their orders, estimated delivery dates, and real-time updates.
2. Logistics and transportation: Logistics and transportation companies can use package tracking bots to streamline their operations. They can automate the tracking process and provide real-time updates to customers and stakeholders.
3. Retail: Retail businesses can also benefit from package tracking bots. They can track their shipments and inventory in real-time, and provide customers with a seamless delivery experience.
4. Healthcare: Package tracking bots can also be used in healthcare to track the delivery of medical supplies and equipment. Hospitals, clinics, and pharmacies can use this technology to ensure that they have the supplies they need, when they need them.
5. Travel: Package tracking bots can be used by travel companies to help customers track their luggage and other belongings while traveling. This can help reduce lost luggage and improve the customer experience.

**9. Working / Code**

User message/ track command (request)

The ‘views.py’ file in ‘Order Tracking Bot’ receives the request. Depending upon the request views.py may send the following replies:

1.If hi \*goto step 2.5\* 2.if ‘track’ in user message \*goto step 3\*

2.5\* The bot reply with “Welcome to OrderTrackingBoB!!! To track your package, please include 'track' in your message and end the message with the tracking id provided by Ekart. For example: hello track 'tracking id”

This acts as a command to track an order. The user details such as first name, last name, tracking\_id, chat\_id and creation date is store in the database.

The views.py then converts the message into a list and then picks the last element as the tracking id.

A function ‘processed\_data’ is then called from the driver code, which in turns calls another code from the same code that fetches the information of the specified tracking id in the form of dictionary and gives it to ‘processed\_data’. The function then processes the data and sends the related data to views.py as reply text.

The views.py then simply forwards this reply text to the user with all the necessary information.

The Bot then awaits for a new message.

**Code**:

1) Main Communication Handling code / Driving Code:

def home(request):

    try:

        body = json.loads(request.body)

        print(body)

        reply\_text = f"Hi {body.get('message').get('from').get('first\_name')}!\nWelcome to OrderTrackingBoB!!!\nTo track your package, please include 'track' in your message and end the message with the tracking id provided by Ekart.\nFor example: hello track 'tracking id'"

        user\_text = body.get('message').get('text')

        chat\_id = body.get('message').get('from').get('id')

        m\_reply\_id = body.get('message').get('message\_id')

        lst = user\_text.split(' ')

        tracking\_id = lst[-1]

        if 'track' in user\_text.lower():

            user\_obj = UserInfo(

                chat\_id = chat\_id,

                first\_name = body.get('message').get('from').get('first\_name'),

                last\_name = body.get('message').get('from').get('last\_name'),

                tracking\_id = tracking\_id

            )

            try:

                user\_obj.save()

            except:

                reply\_text = 'You are already subscribed to this tracking id!'

        # if '/start' in user\_text.lower():

            reply\_text = track\_send.processed\_data(tracking\_id)

        Bot.send\_message(chat\_id=chat\_id, text=reply\_text, parse\_mode=telegram.ParseMode.MARKDOWN, reply\_to\_message\_id=m\_reply\_id)

    except:

        print(traceback.format\_exc())

        Bot.send\_message(chat\_id=chat\_id, text='hi, error')

    return HttpResponse('hi')

2) Data Fetching Function:

def data\_retriver(tracking\_id):

    url = "https://ekartlogistics.com/ws/getTrackingDetails"

    payload = json.dumps({

      "trackingId": tracking\_id

    })

    headers = {

      'Content-Type': 'application/json'

    }

    response = requests.request("POST", url, headers=headers, data=payload)

    data = response.json()

    print(type(data))

    print(data)

    return data

def processed\_data(tracking\_id):

  try:

    data = data\_retriver(tracking\_id)

    print(type(data),data)

    exp\_date = data.get('expectedDeliveryDate')

    exp\_date = exp\_date/1000

    expected\_date = str(datetime.date.fromtimestamp(exp\_date))

    print('\n\\\\',exp\_date,expected\_date)

    data\_list = list(data['shipmentTrackingDetails'])

    print('@@@',type(data\_list),data\_list,'@@@')

    last\_data = data\_list[-1]

    latest\_date = last\_data['date']/1000

    date = str(datetime.datetime.fromtimestamp(latest\_date))

    merchant = data['merchantName']

    reply\_text = f"Latest update for your package from {merchant} is:-\n\nLocation:- {last\_data['city']}\nLast updated on:- {date}\nExpected date:- {expected\_date}\nStatus Update:- {last\_data['statusDetails']}"

  except:

     reply\_text = 'Error! Could not fetch data!'

     print(traceback.format\_exc())

  return reply\_text

sample data retrieved:

The data is in the form of ‘dictionary’.

{'shipmentType': 'Forward', 'expectedDeliveryDate': 1682087890000, 'receiverName': 'ShaanRawat', 'receiverRelationShip': None,

'merchantName': 'flipkart.com', 'sourceCity': 'JAIPUR', 'destinationCity': 'Gurgaon', 'reachedNearestHub': True, 'faShipment': False,

'shipmentTrackingDetails': [{'date': 1681868973000, 'city': 'JAIPUR', 'statusDetails': 'Shipment Created'},

{'date': 1681905458000, 'city': 'JAIPUR', 'statusDetails': 'Pickup From Seller'}, {'date': 1681936437000, 'city': 'JAIPUR', 'statusDetails': 'Received at FKL\_Jaipur\_BTS'},

{'date': 1681868976000, 'city': 'Jaipur', 'statusDetails': 'Dispatched to JAI/BTS'}, {'date': 1681936441000, 'city': 'Jaipur', 'statusDetails': 'Received at JAI/BTS'},

{'date': 1681937136000, 'city': 'Jaipur', 'statusDetails': 'Received at JAI/BTS'}, {'date': 1681975155000, 'city': 'New Delhi', 'statusDetails': 'Dispatched to Bamnoli Sort Centre'},

{'date': 1682011250000, 'city': 'New Delhi', 'statusDetails': 'Received at Bamnoli Sort Centre'}, {'date': 1682025970000, 'city': 'Gurgaon', 'statusDetails': 'Dispatched to Badshahpur Hub'},

{'date': 1682037452000, 'city': 'Gurgaon', 'statusDetails': 'Received at Badshahpur Hub'}, {'date': 1682043362000, 'city': 'Gurgaon', 'statusDetails': 'Out For Delivery'},

{'date': 1682087890000, 'city': 'Gurgaon', 'statusDetails': 'Delivered'}]}

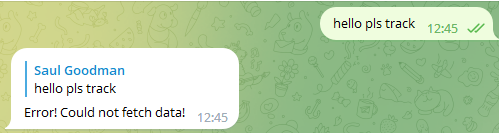
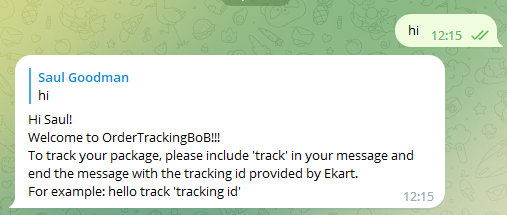
**10. Input Data and Validation of the Project**

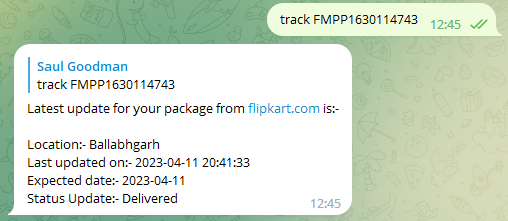
Input Data:

* User input messages containing tracking IDs and commands
* User details such as name and chat ID
* Order and package details such as tracking number, courier name, and delivery status
* Database containing user and order details

Validation:

* Validate user input messages for correct syntax and format (e.g. tracking ID is included and in correct format)
* Validate user details (e.g. chat ID is valid)
* Validate order and package details (e.g. tracking number is valid and matches courier)
* Ensure data is securely stored and accessed only by authorized personnel
* Implement error handling and reporting mechanisms for any invalid or incorrect data





**12. Project Vision**

Our project vision is to create a user-friendly and efficient package tracking bot that provides customers with real-time updates on the status and location of their packages. Our bot will streamline the package tracking process, making it faster and more convenient for customers to receive updates on their shipments. By implementing innovative technologies and a user-centered approach, our bot will enhance the customer experience and improve satisfaction with the package tracking process.

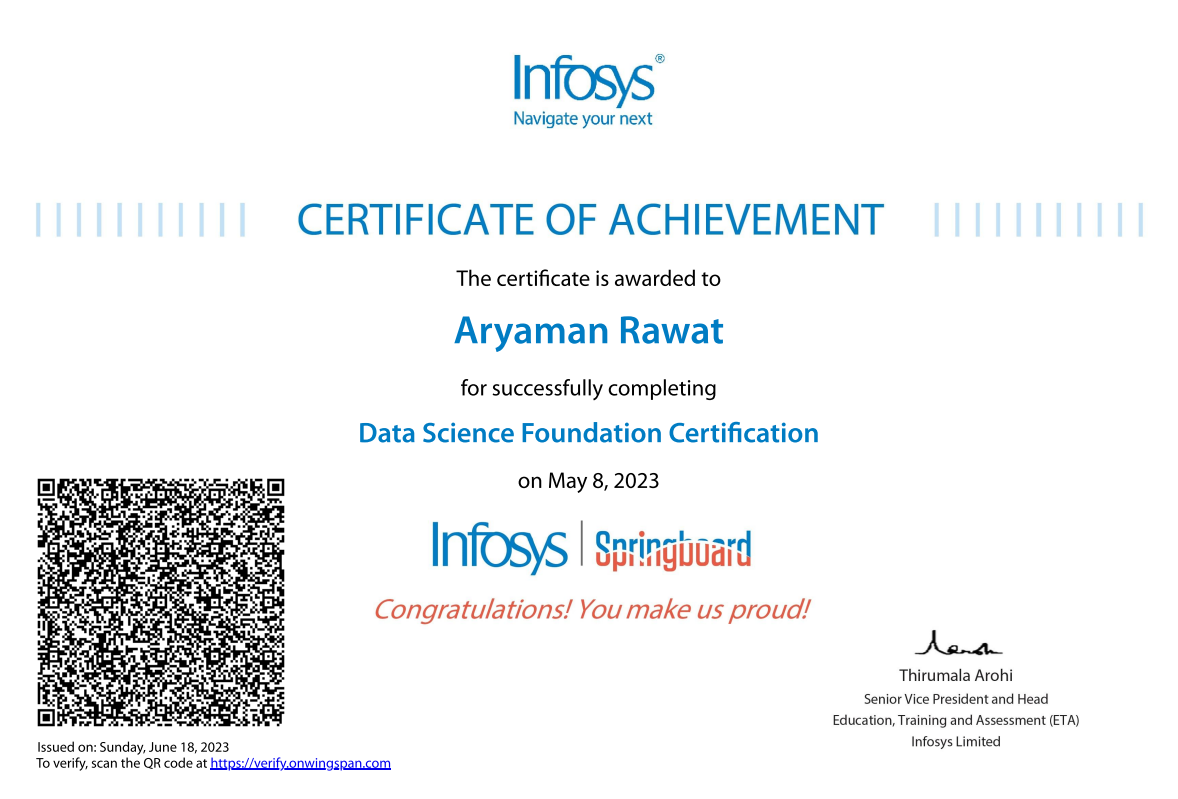
**14. Conclusion**

In conclusion, a package tracking bot can provide a convenient and efficient way for users to track their packages without the need for manual input and monitoring. By using the Telegram platform, the bot [1], [2]can reach a wide audience and provide real-time updates on package delivery status. The use of database storage can also allow for personalized tracking information for each user. Overall, a package tracking bot can improve the user experience and streamline the package tracking process.[3]

**CERTIFICATE**

****

****

****