**1. INTRODUCTION**

**Chatbot** is the cloud service who can access from anywhere and any one access this bot if they register in integrated messenger.

The main objective of this project is to create bot on “Aditya” at which we’ll get all the information about this college.

If we need any information about fees or about placements, no need to contact with higher authorities. We can know the information according to our use. Simply we need to interact with some Third-parties in which we are integrated like Facebook, web Demo, telegram, and some other similar messengers. Then start conversion with that messenger with which you are connected. That messenger will give response in fraction of seconds.

By using this bot, we save the work burden on management and on users . This bot can use by faculty, students ,parents and any others to get information about our College.

* 1. **EXISTING SYSTEM**

In existing system, if user need any information about buses, they have to contact Transport department and then they need any information related college fees, about placements, faculty, department and other similar information about this college they need to meet respective departments.

***Problems in the Existing System***

User may not get the exact information about the college by contacting only one person. The process of contacting the required person was also not very easy to get information related to this college and it will take much time.

**1.2 PROPOSED SYSTEM**

The user can simply interact with the messenger which we are integrated then the bot will give response immediately by replaying for the user’s query.

***Advantages of the proposed system***

1. It reduces the time.
2. No need to contact with the manage ment for small information.
3. It reduces the work burden.
4. It provides an easy, systematic and maintained environment for the management.

**2. REQUIREMENT ANALYSIS**

**2.1 SYSTEM REQUIREMENT SPECIFICATION**

This Project is a mobile application. The hardware and software specifications of the system are mentioned below.

* + 1. ***Hardware Requirements***
* Processor : Intel Pentium-IV and above
* Hard disk : 500GB Min.
* RAM : 2GB Min. & Above
* Others : If any Applicable

***2.1.2 Software Requirements***

* Database : Cloud
* Design : API

**2.2 SOFTWARE REQUIREMENT SPECIFICATION**

***2.2.1 Vision***

1. Nowadays the problem with this system is to contact with the management and to ask something about this college. If it is not related to the information what they need, then it is time waste process.
2. This system was designed to interact with the user, and this user can be faculty or student or any employee or any user.
3. If faculty need any information about the any department, no need to contact with the respective department. They simply interact with the integrated messenger and chat with the bot and it will give the response related to the user’s query.
4. If student/others need any information about this college, about the departments, or about labs, about fees structure for required department, and about faculty, they will get quick response from the bot.

***2.2.2 Glossary***

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Students/Others** | Those who need information about this college. |
| **Developer/Reviewer** | A person who views the history records and modify the existing details if any. |

***Table 2.1*** Glossary

***2.2.3 Scope***

The Chatbot aims at helping the user to give the information about our college and interact with the user.

The major benefits of this services are-

* It is a unique service which helps to organize event without any work with burden.
* It has a wide variety of modules.

By just few clicks user can know the details about this college. The users no need to get hurry to get the information related to this college. Not only for students, this bot can also use for the people who need to join in this college and also for faculty.

***2.2.4 System Functions***

The following list in tabular format as shown.

* Registration of Users:

Any student who want the information about this college through this bot

Login in integrated messenger

Connect with the bot

Ask any queries about this college

* Developer

Login

Add the intents which are not there in this intent previously

View the history

Add zip files and intents if any

Logout

* Reviewer

Login

View the intents and history

Logout

|  |  |  |
| --- | --- | --- |
| **S.No** | **System Functions** | **Description** |
|  | S1 | **User Login** |
| **1** | S1.1 | Login with the messenger in which we are integrated |
| **2** | S1.2 | Ask the query about this college. |
|  | S2 | **Developer Login** |
| **3** | S2.1 | The developer will login in the bot and view the intents and add zip files if any. |
|  | S2.2 | Developer will view all the history and he/she will know that how much the user will spend time with this bot. |
| **4** | S3 | **Reviewer Login** |
|  | S3.1 | The reviewer will login in the bot and view the intents and history. |
|  | S3.2 | Developer will view all the history and he/she will know that how much the user will spend time with this bot. |

***Table 2.2*** System Functions

***2.2.5 Detailed Software Requirement***

**Actors**

|  |  |
| --- | --- |
| **Actor Name** | **Hide App** |
| **Primary actor** | User |
| **Pre-condition** | User successfully logs in and connect with this bot. |
| **Main Scenario** | User after successfully connect, then he/she can ask any queries related to this college. |

***Table 2.3*** Use Case Model for Student

|  |  |
| --- | --- |
| **Actor Name** | **System authorization** |
| **Primary Actor** | Developer |
| **Pre-condition** | Should be pre-registered as developer. |
| **Main scenario** | 1. Login with his credentials. 2. Developer train the bot to respond to the user. 3. Views all the history. 4. Add the intent if any. |
| **Alternate scenario** | 1. Advice the user, if he/she ask any unrelated query about this college. 2. And then allow the user to re-enter the query. |

***Table 2.4*** Use Case Model for Developer

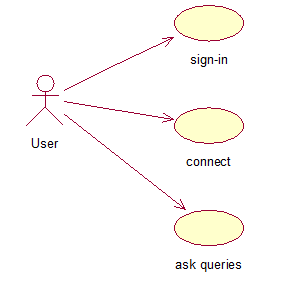
|  |  |
| --- | --- |
| **Actor Name** | **System authorization** |
| **Primary Actor** | Reviewer |
| **Pre-condition** | Should be pre-registered as reviewer added by developer. |
| **Main scenario** | 1. Login with his credentials.  2. Developer train the bot to respond to the user.   1. Views all the history. |
| **Alternate scenario** | 1. Advice the user, if he/she ask any unrelated query about this college. 2. And then allow the user to re-enter the query. |

***Table 2.5*** Use Case Model for Reviewer

**List of Use Cases**

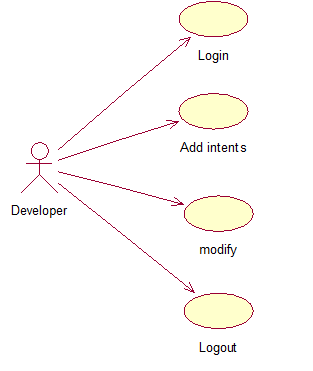
* Use Case Diagram for User
* Use Case Diagram for Developer
* Use Case Diagram for Reviewer

**Use Case Diagrams**

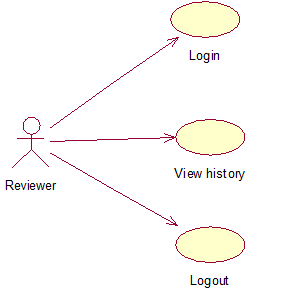


***Fig 2.6:*** Use Case Diagram for User

User need to sign-in with the integrated messenger, connect with the bot and then gather the information by ask the queries.

 ***Fig 2.7*** Use Case Diagram for Developer

Developer need to login in the bot, add the intents if any new queries related to the college, modify the bot by adding new zip files if require and logout.



***Fig 2.8*** Use Case Diagram for Reviewer

Reviewer need to login in the bot, view the history and logout.

***2.2.6 Detail Use case Description***

|  |  |
| --- | --- |
| **Use Case Name** | **User** |
| Use Case ID | UC 1 |
| Actor(s) | User |
| Goal | Student connect with the bot and receive response from the bot. |
| Summary | User need to connect with the bot and ask any queries related to this college. |
| Preconditions |  |
| Main Flow | |  |  | | --- | --- | | 1. The user enters the use case. 2. User selects the option of application:S1 3. This tool would perform the respective action.S2 | * 1. System displays the the option of the application bot to ask queries by the user.   S1 Selected option is executed.  S2 This tool would do its respective task. | |
| Sub Flows | **S1: User registers into the bot**   |  |  | | --- | --- | | 1.User selects integrated messenger | 1.1 User after selecting the messenger. They need to connect with the messenger and then ask any queries. |   **S2: Action performed by tool**   |  |  | | --- | --- | | 1.S2 | 1.1 After User successfully asks queries related to the college, then he/she will get response based based on the query | |
| Alternate Flows | **-** |
| Post Conditions | **-** |
| Cross References | **-** |
| Assumptions | **-** |
| Business Rules | 1. User should ask query related to the college.   2. The User should be authenticated with the response. |

***Table 2.6*** Use Case Description for User

|  |  |
| --- | --- |
| **Use Case Name** | **Admin** |
| Use Case ID | UC 2 |
| Actor | Developer |
| Goal | To do modifications to the bot if any. |
| Summary | Receives the history of the bot. |
| preconditions |  |
| Main Flow | |  |  | | --- | --- | | 1. The Developer enters the use case.  2. Developer can do anyone or both options: S1, S2 | 1.1 The Developer logs into the portal.  S1. Add the intents i.e.., user’s expression which is not in the current intent.  S2. View the history of the bot.  2.1 Selected option is executed. | |
| Sub Flows | **S1: Developer adds user’s expression**   |  |  | | --- | --- | | 1.Developer selects S1 | 1.1 Developer will add the intents to the current one. |   **S2: Developer views the history**   |  |  | | --- | --- | | 1.Developer selects S2 | 1.1 Developer will view the history of the bot. | |
| Alternate Flows | - |
| Post Conditions | - |
| Cross References | - |
| Assumptions | - |
| Business Rules | Developer will view the history, if he/she need to add to intent or not. |

***Table 2.7*** Use Case Description for Developer

|  |  |
| --- | --- |
| **Use Case Name** | **Admin** |
| Use Case ID | UC 3 |
| Actor | Reviewer |
| Goal | To view the modifications to the bot if any. |
| Summary | Receives the history of the bot. |
| preconditions |  |
| Main Flow | |  |  | | --- | --- | | 1. The Reviewer enters the use case.  2. Reviewer can do the options: S1. | 1.1 The Reviewer logs into the portal.  S1. View the history of the bot and modifications.  2.1 Selected option is executed. | |
| Sub Flows | **S1: Developer adds user’s expression**   |  |  | | --- | --- | | 1.Reviewer selects S1 | 1.1 Reviewer views the history and modifications will add the intents to the current one. | |
| Alternate Flows | - |
| Post Conditions | - |
| Cross References | - |
| Assumptions | - |
| Business Rules | Reviewer will view the history and modifications if any. |

***Table 2.8*** UseCase Description for Reviewer

***2.2.7 Functional Capabilities***

**Login of the users:**

The user need to sign-in with integrated messenger and then connect with that messenger.

**Asking queries about college:**

After connected with the integrated messenger, he/she may ask any queries related to this college.

**Response:**

Users will get quick response related to the college.

***2.2.8 Business Rules / Validations***

* Queries related to the college should be valid.

**3. SYSTEM DESGIN**

SYSTEM DESIGN phase follows analysis phase. Design is maintaining record of proof design divisions and providing a blueprint for the implementation phase. Design is the bridge between system analysis and system implementation.

System design is transition from a user oriented, document oriented to programmers. The design is a solution, a “how to” approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through logical and physical stages of development; a logical design reviews the present physical system, prepare input and an output specification, detail the implementation plan, and prepares a logical design walkthrough.

**Design Methodology**

The design process for software system has two levels:

1. System design or Top-level design
2. Detailed design or Logical design.

**System Design**

In the system design the focus is on deciding which modules are needed for the system, the specification of these modules and how these modules should be interconnected.

**Detailed Design**

In detailed design, the interconnection of the modules or how the specification of the modules can be satisfied is decided. Some properties for a software system design are

1. Verifiability
2. Completeness
3. Consistency
4. Traceability
5. Simplicity / Understandability

**3.1 DATA DICTIONARY**

After carefully understanding the requirements of the client the entire data storage requirements are divided into tables.

But in this service, no need of create any databases because the code will generate automatically in the background in JSON format.

**3.2 LOGICAL DATABASE DESIGN**

***3.2.1 Normalization***

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity.

**Insertion anomaly**  Inability to add data to the database due to absence of other data.

**Deletion anomaly** Unintended loss of data due to deletion of other data.

**Update anomaly** Data inconsistency resulting from data redundancy and partial update.

***3.2.2 Normal Forms***

These are the rules for structuring relations that eliminate anomalies.

**First Normal Form**

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

**Second Normal Form**

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1. Primary key is a not a composite primary key.

2. No non-key attributes are present

**Third Normal Form**

A relation is said to be in third normal form if their exits no transitive dependencies.

***Transitive Dependency***

If two non-key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

**3.3 UML DIAGRAMS**

Unified Modeling Language is the language used to visualize, specify, construct and document any component of software engineering.

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly unique perspective. Each view is defined by a set of diagrams, which is as follows.

**User Model View**

1. This view represents the system from the user’s perspective.
2. The analysis representation describes a usage scenario from the end-user’s perspective.

**Structural model view**

1. In this model, the data and functionality are arrived from inside the system.
2. This model view models the static structures.

**Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

**Environmental Model View**

In this structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are

1. UML analysis modeling, which focuses on the user model and structural model views of the system.
2. UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Every complex system is best approached through a small set of nearly independent views of a model; no single viewer is sufficient. Every model may be expressed at different levels of fidelity. The best models are connected to reality.

The UML includes nine such diagrams:

**Static Diagrams includes**

* Class diagram
* Object diagram
* Component diagram
* Deployment diagram

**Dynamic diagrams include**

* Use case diagram
* Sequence diagram
* Collaboration diagram
* State chart diagram
* Activity diagram

**Class diagram**

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system.

**Object diagram**

An object diagram shows a set of objects and their relationships. Object diagrams represent static snapshots of instances of the things found in class diagrams. These diagrams address the static design view or static process view of a system as do class diagrams, but from the perspective of real or prototypical cases.

**Component diagram**

A component diagram shows the organizations and dependencies among a set of components. Component diagrams address the static implementation view of a system. They are related to class diagrams in that a component typically maps to one or more classes, interfaces, or collaborations.

**Deployment diagram**

A deployment diagram shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of an architecture. They are related to component diagrams in that a node typically encloses one or more components.

**Use case diagram**

A use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system.

**Sequence diagram**

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. Sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specifications.

**Collaboration diagram**

Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

**State chart diagram**

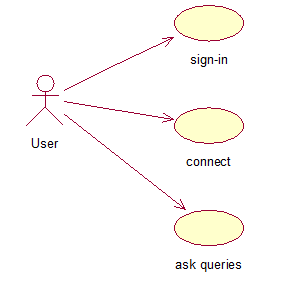
A state chart diagram shows a state machine, consisting of states, transitions, events, and activities. State chart diagrams address the dynamic view of a system. They are especially important in modeling the behavior of an interface, class, or collaboration and emphasize the event-ordered behavior of an object, which is especially useful in modeling reactive systems.

**Activity diagram**

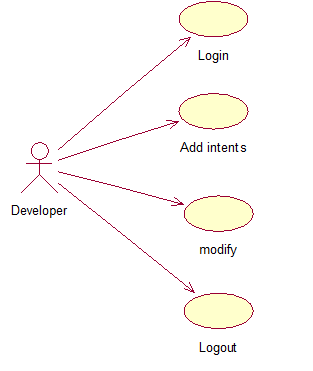
An activity diagram is a special kind of a state chart diagram that shows the flow from activity to activity within a system. Activity diagrams address the dynamic view of a system. They are especially important in modelling the function of a system and emphasize the flow of control among objects.

***3.3.1 Use Case Diagram***

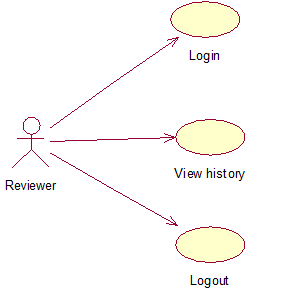
**U**se case diagrams are created to visualize the relationships between actors and use cases. A use case is a pattern of behavior the system exhibits. Each use case is a sequence of related transactions performed by an actor and the system.



***Fig 3.1***Use case diagram for user



***Fig 3.2*** Use Case Diagram for Developer



***Fig 3.3*** Use Case Diagram for Reviewer

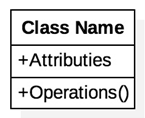
***3.3.2 Class Diagram***

A Class diagrams describe the static structure of a system, or how it is structured rather than how it behaves. Class diagram gives an overview of a system by showing its classes and the relationships among them. UML class is a rectangle divided into: class name, attributes, and operations. Our class diagram has three kinds of relationships.

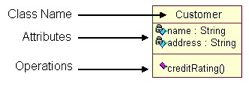
* **Association** -- A relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other to perform its work.
* **Aggregation** -- an association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole. In our diagram, **Order** has a collection of **Order Details**.
* **Generalization** -- an inheritance link indicating one class is a super class of the other.

Class diagrams are widely used to describe the types of objects in a system and their relationships.  Class diagrams model class structure and contents using design elements such as classes, packages and objects.  Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design.  This example is only meant as an introduction to the UML and class diagrams.  If you would like to learn more see the [Resources](http://atlas.kennesaw.edu/%257Edbraun/csis4650/A&D/UML_tutorial/resources.htm) page for more detailed resources on UML.

Classes are composed of three things: a name, attributes, and operations.  Below is an example of a class.

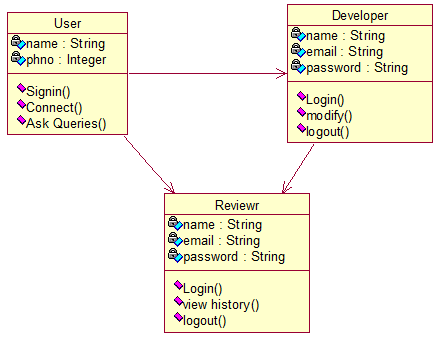


***Fig 3.4*** Class Icon

**

***Fig 3.5*** Sample Class Diagram

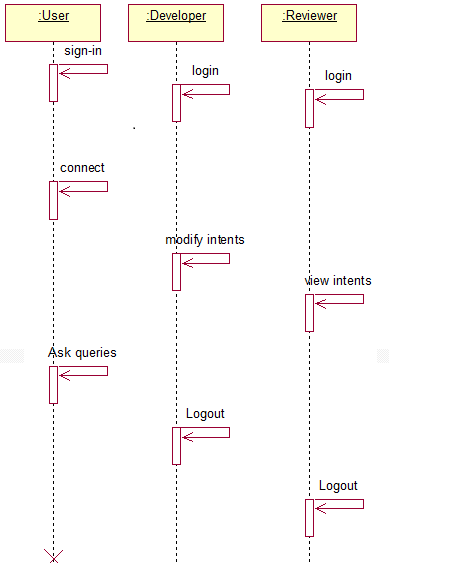
Class diagrams are the mainstay of object-oriented analysis and design. UML class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling.



***Fig 3.6*** Class Diagram User, Developer and Reviewer

***3.3.2 Interaction Diagram***

A type of interaction diagram, a sequence diagram shows the actors of the object participating in an interaction and the events they generate arranged in a time sequence. Often a sequence diagram shows the events that results from a instance of a particular instance of a use case but a sequence diagram can also exist in a more generic form. The vertical dimension in a sequence diagram represents time, with time proceeding down the page the horizontal dimension represents different actors.



***Fig 3.7*** Sequence Diagram for User, Developer, Reviewer

**Collaboration Diagram**



***Fig 3.8*** Collaboration of User

**4.MODULES**

**4.1 MODULE DESCRIPTION**

Local File Transfer System (crate) includes 4 basic modules. They are

1. User
2. Developer
   * 1. ***User***
3. **Login**

User need to sign-in with the integrated messenger and then connect with the bot.

1. **Asking queries**

User should ask queries related to the college and these queries should be valid and necessary his/her need.

***4.1.2 Developer***

1. **Login:**

Developer must login with his userid and password. If it is not correcting then need to contact cloud services.

1. **Add Intents:**

After Developer login to his/her portal, he/she may add intents and zip files if there is any need. And can-do modifications is it needs.

1. **View History:**

The developer can view history and add the related queries to the respective intent asked by the user but not in the intent.

**5. SYSTEM IMPLEMENTATION**

**5.1 SELECTED SOFTWARES**

***5.1.1*** **Software’s Interfaces:**

All databases for the OLMS will be configured using MySQL. These databases include student information. These can be modified by the end users. The student database will include the name, registration no, email-id, and their respective password.

And the e-form consists of registration no, Type of leave, Subject, Number of days, Reason for Leave.

To implement the project, we have chosen HTML language for its more interactive and easy to understand support.

**Multiple language support**: Android supports multiple languages.

**Communication Interfaces:**

This System supports Google chrome and Mozilla Firefox web browser.

**Programming Environment**:

MYSQL:

MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS); in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source [client–server model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) RDBMS.

My The [SQL](https://en.wikipedia.org/wiki/SQL) acronym stands for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). [Free-software](https://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL. Applications that use the MySQL database.

CASCADING STYLESHEETS:

Cascading Style Sheets (CSS) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language). Although most often used to set the visual style of [web pages](https://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](https://en.wikipedia.org/wiki/HTML) and [XHTML](https://en.wikipedia.org/wiki/XHTML), the language can be applied to any [XML](https://en.wikipedia.org/wiki/XML) document.

CSS is designed primarily to enable [the separation of document content from document presentation](https://en.wikipedia.org/wiki/Separation_of_presentation_and_content), including aspects such as the [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation

Characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

HTML:

Htmlis a [markup language](https://en.wikipedia.org/wiki/Markup_language) used for structuring and presenting content on the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). It was finalized, and published, on 28 October 2014 by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C).This is the fifth revision of the [HTML](https://en.wikipedia.org/wiki/HTML) standard since the inception of the World Wide Web. The previous version, HTML 4, was standardized in 1997.

Its core aims are to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices

**Additional hardware support :**

The system shall run on:

Operating system: Any Windows OS.

Scripts which supports HTML, CSS, JavaScript.

Web Browser: Google Chrome version 33.0 or higher

**5.2 SAMPLE CODE:**

***About IT: -***

***Placements: -***

{

"user Says": [

{

"id": "2c398a1f-7c0a-4176-b520-4f688ff765e6",

"data": [

{

"text": "placements for information technology department"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "3a753db9-7ef1-4c64-a430-5b79a75397bb",

"data": [

{

"text": "information technology department placements"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "dcad47c4-99b6-4c9f-8ae4-4c4c7030e623",

"data": [

{

"text": "placements for information technology"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "1c2e3ae2-b986-45fd-8172-b45ef5c2b8b7",

"data": [

{

"text": "placements for IT"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "92fcfb3a-ee3c-44b4-9a17-4ceb67804f6f",

"data": [

{

"text": "information technology placements"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "96d30eb5-d0a4-4729-b2b9-9d746c2b949a",

"data": [

{

"text": "IT placements"

}

],

"isTemplate": false,

"count": 0

}

],

"id": "1f2232b0-70e0-461c-bb4a-dab44e4654ce",

"name": "IT placements",

"auto": true,

"contexts": [],

"responses": [

{

"reset Contexts": false,

"affectedContexts": [],

"parameters": [],

"messages": [

{

"type": 3,

"platform": "telegram",

"image URL": "http://www.crescendoworldwide.org/wp-content/uploads/2017/01/4-effective-tips-to-a-success-business-process-outsourcing-strategy-624x374.jpg"

},

{

"type": 0,

"platform": "telegram",

"speech": "Companies Visiting our campus:\nTata Consultancy Services\Wipro\Infosys\nHCL Technologies etc.....\n No.of students placed:220"

},

{

"type": 2,

"platform": "telegram",

"title": "You May Like",

"replies": [

"IT jobs",

"IT staff",

"IT labs"

]

},

{

"type": 3,

"platform": "facebook",

"imageUrl": "http://www.crescendoworldwide.org/wp-content/uploads/2017/01/4-effective-tips-to-a-success-business-process-outsourcing-strategy-624x374.jpg"

},

{

"type": 0,

"platform": "facebook",

"speech": "Companies Visiting our campus:\nTata Consultancy Services\nWIPRO\nInfosys\nHCL Technologies etc......\n No.of students placed:220"

},

{

"type": 2,

"platform": "facebook",

"title": "You May Like",

"replies": [

"IT jobs",

"IT placements",

"IT staff"

]

},

{

"type": 0,

"speech": "Companies Visiting our campus:\nTata Consultancy Services\nWIPRO\nInfosys\nHCL Technologies etc...... \n No.of students placed:220. For more information: http://aec.edu.in/?p\u003dPlacements"

}

]

}

],

"priority": 500000,

"webhookUsed": false,

"webhookForSlotFilling": false,

"fallbackIntent": false,

"events": []

}

***Staff:-***

{

"userSays": [

{

"id": "09bdaa85-c6c8-4cf7-b5ef-98e0f511ae24",

"data": [

{

"text": "staff for information technology department"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "ec40ada8-121d-4451-a36a-24d4fb7a56fe",

"data": [

{

"text": "staff for information technology"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "d030e01a-9ecb-476e-a776-9a5d0a2aa6e3",

"data": [

{

"text": "information technology department staff"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "eae77cf9-58c0-420c-83fb-22ed759459f8",

"data": [

{

"text": "information technology staff"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "34463507-844a-46b3-828c-f21ee96e7f9f",

"data": [

{

"text": "staff for IT"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "81d02d66-bc18-4fae-b4f9-bdf0f10b7f5b",

"data": [

{

"text": "It staff"

}

],

"isTemplate": false,

"count": 0

}

],

"id": "2d95c7f9-4471-481c-8dbf-48157d297648",

"name": "IT staff",

"auto": true,

"contexts": [],

"responses": [

{

"resetContexts": false,

"affectedContexts": [],

"parameters": [],

"messages": [

{

"type": 3,

"platform": "telegram",

"imageUrl": "http://jntuk-coeerd.in/fdpimages/aditya.jpg"

},

{

"type": 0,

"platform": "telegram",

"speech": "Faculty\n\nThe vibrant faculty members of the department possess demonstrated expertise in many areas of information technology and flair for teaching different courses. The department consists of a medley of faculty members with industrial and academic experience. Faculty members are involved in carrying out collaborative research projects with leading industries and international universities. In addition to the above, the faculty membersâ€™ possess essential attributes of being a good mentor to students."

},

{

"type": 2,

"platform": "telegram",

"title": "you may like",

"replies": [

"IT labs",

"IT jobs",

"IT placements"

]

},

{

"type": 3,

"platform": "facebook",

"imageUrl": "http://jntuk-coeerd.in/fdpimages/aditya.jpg"

},

{

"type": 0,

"platform": "facebook",

"speech": "Faculty\n\nThe vibrant faculty members of the department possess demonstrated expertise in many areas of information technology and flair for teaching different courses. The department consists of a medley of faculty members with industrial and academic experience. Faculty members are involved in carrying out collaborative research projects with leading industries and international universities. In addition to the above, the faculty membersâ€™ possess essential attributes of being a good mentor to students."

},

{

"type": 2,

"platform": "facebook",

"title": "you May Like",

"replies": [

"IT labs",

"IT jobs",

"IT placements"

]

},

{

"type": 0,

"speech": "Faculty\n\nThe vibrant faculty members of the department possess demonstrated expertise in many areas of information technology and flair for teaching different courses. The department consists of a medley of faculty members with industrial and academic experience. Faculty members are involved in carrying out collaborative research projects with leading industries and international universities. In addition to the above, the faculty membersâ€™ possess essential attributes of being a good mentor to students."

}

]

}

],

"priority": 500000,

"webhookUsed": false,

"webhookForSlotFilling": false,

"fallbackIntent": false,

"events": []

}

***IT:-***

{

"userSays": [

{

"id": "01c048d9-4c3e-4630-873a-7f5956f493e6",

"data": [

{

"text": "it hod name"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "1021867d-c1fa-419b-a673-9648fc119257",

"data": [

{

"text": "IT hod"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "c812e1ae-bf25-405b-a224-0c43a181c000",

"data": [

{

"text": "who is it hod"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "c24048b0-6bf9-4356-b639-94f0bff2bd78",

"data": [

{

"text": "tell me about information technology department"

}

],

"isTemplate": false,

"count": 1

},

{

"id": "1a4e7faa-5411-4c1f-9ce2-a9cb40b47a34",

"data": [

{

"text": "tell me about information technology"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "14c38a3d-0df9-4acb-8ac3-979efb5dbdb5",

"data": [

{

"text": "tell me about IT"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "a190e873-ecb9-4288-b053-0d120935d9b7",

"data": [

{

"text": "information technology department"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "53583e2e-f091-40b6-92d8-a18808364613",

"data": [

{

"text": "information technology"

}

],

"isTemplate": false,

"count": 0

},

{

"id": "c397fa01-f4cd-4a55-baba-e4a0ef2233c1",

"data": [

{

"text": "IT"

}

],

"isTemplate": false,

"count": 1

}

],

"id": "49179e8e-b0b1-47dc-9e41-bf7731716a35",

"name": "IT",

"auto": true,

"contexts": [],

"responses": [

{

"resetContexts": false,

"affectedContexts": [],

"parameters": [],

"messages": [

{

"type": 3,

"platform": "telegram",

"imageUrl": "http://www.hit.ac.in/img/slides/Information%20Technology%20Eng.jpg"

},

{

"type": 0,

"platform": "telegram",

"speech": "VISION and MISSION OF THE DEPARTMENT OF IT\n\nThe mission of the department of Information Technology @ Aditya is to provide quality and value-laden education for students, in the traditional and contemporary areas of information technology and develop students with good ethical practices, thus showcasing their all-round personality development.\n no.of seats :60"

},

{

"type": 2,

"platform": "telegram",

"title": "You May Like",

"replies": [

"IT labs",

"IT jobs",

"IT placements",

"IT staff"

]

},

{

"type": 3,

"platform": "facebook",

"imageUrl": "http://www.hit.ac.in/img/slides/Information%20Technology%20Eng.jpg"

},

{

"type": 0,

"platform": "facebook",

"speech": "VISION and MISSION OF THE DEPARTMENT OF IT\n\nThe mission of the department of Information Technology @ Aditya is to provide quality and value-laden education for students, in the traditional and contemporary areas of information technology and develop students with good ethical practices, thus showcasing their all-round personality development.\n no.of seats :60"

},

{

"type": 2,

"platform": "facebook",

"title": "You May Like",

"replies": [

"IT labs",

"IT staff",

"IT jobs",

"IT placements"

]

},

{

"type": 0,

"speech": "Name of HOD: M.Raja Babu garu,\n\nNo.of seats:60,\n\nNo.of staff:15,\n\nNo.of Labs:3,\n\nNo.of Students Placed:220."

}

],

"defaultResponsePlatforms": [

"google"

]

}

],

"priority": 500000,

"webhookUsed": false,

"webhookForSlotFilling": false,

"fallbackIntent": false,

"events": []

}

**6. TEST CASES**

**6.1 SOFTWARE TESTING**

In any software development, testing is a process to show the correctness of program and it needs the design specifications. Testing is needed to prove correctness completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary to ensure completeness of testing, improve the quality of software and reduce the testing costs and to reduce study needs and operation time.

**6.2 GOALS OF TESTING**

The following are goals of testing…

1. Testing is a process of executing a program with the intent of finding error.
2. A good test case is the one that has a high probability of finding an as at undiscovered error.
3. A successful test is one that uncovers an as at undiscovered error.
   1. **TESTING METHODOLOGY**
4. ***Black box testing***

Black Box Testing is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

1. ***White box testing***

White Box Testing is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually the developers are involved in the white box testing.

1. ***Gray box testing***

Gray Box Testing is the process in which the combination of black box and white box techniques is use.

**6.4 LEVEL OF TESTING**

1. ***Unit testing***

Individual components are tested to ensure that they operate correctly. Each component is tested independently without other system components.

1. ***System testing***

The sub-systems are integrated to make up the entire system. The testing process is concerned with finding errors, which result from un-anticipated interactions between subsystem components.

1. ***Integration testing***

sometimes global data structures can represent the problems to uncover errors that are associated with interfacing the objective is to make unit test modules and built a program structure that has been detected by design.

1. ***Acceptance testing***

This is the final stage in the testing process before the system is accepted for operational use. Acceptance testing may reveal errors and omissions in the system requirements definition because real data exercises the system in different ways from the test data.

1. ***Regression testing***

Regression testing is actually that helps to ensure changes that don’t introduce unintended behaviour as additional errors. Regression testing may be conducted manually by executing a subset of all test cases or using automated capture play back tools.

**6.5 Unit Testcases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC ID** | **Input** | **Description** | **Expected Results** | **Pass/Fail** |
| TC01 | Username Blank  Password Blank | A blank Username and password given by the Developer. | Please Enter user Name and password. | Fail |
| TC02 | Username Blank  Password filled | A blank Username given by the Developer | Username or Password is incorrect. | Fail |
| TC03 | Username filled  Password Blank | A blank Password  given by the Developer | Enter Correct Password | Fail |
| TC04 | Username filled  Password filled | Username and password filled by the Developer | Contact with cloud services | Pass |
| TC05 | Username filled Password filled | password filled by the Developer | Goes to home page of the Developer | Pass |

***Table6.1*** Developer login

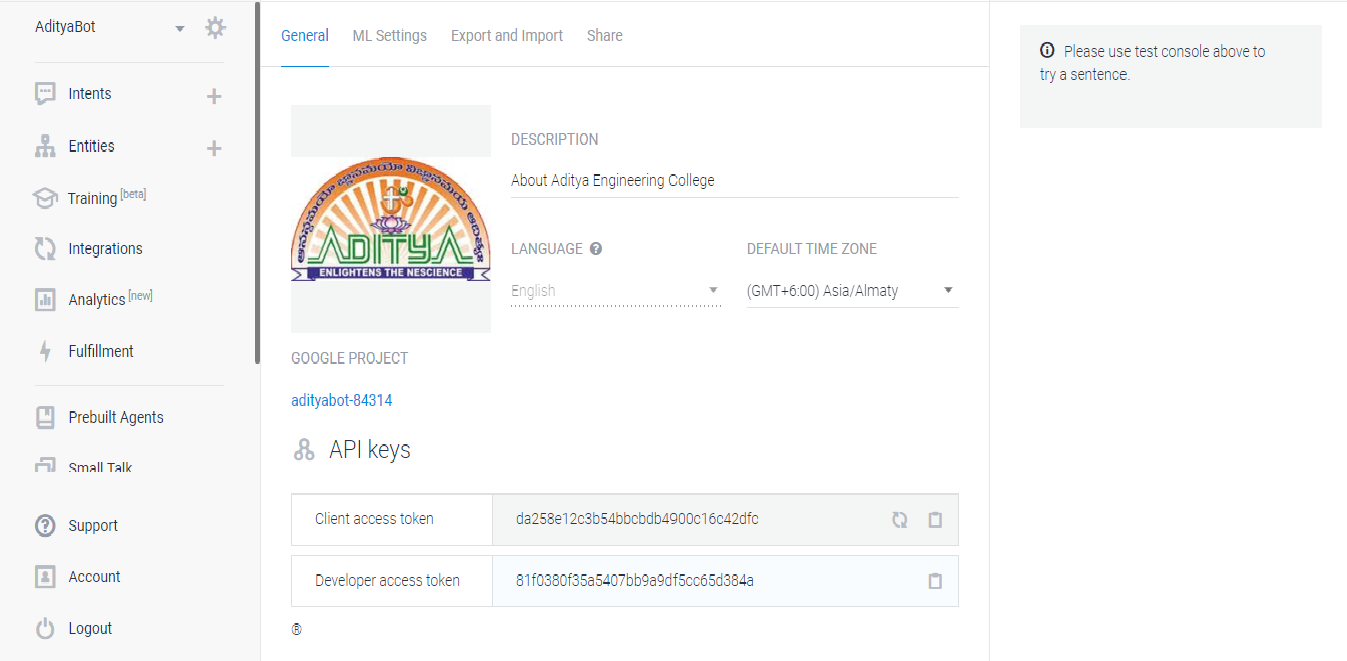
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC ID** | **Input** | **Description** | **Expected Results** | **Pass/Fail** |
| TC01 | Username Blank  Password Blank | A blank Username and password given by the Reviewer. | Please Enter User Name and password. | Fail |
| TC02 | Username Blank  Password filled | A blank Username given by the Reviewer | Username or Password is incorrect. | Fail |
| TC03 | Username filled  Password Blank | A blank Password  given by the Reviewer | Enter Correct Password | Fail |
| TC04 | Username filled  Password filled | Username and password filled by the Reviewer | Contact with cloud services | Pass |
| TC05 | Username filled Password filled | password filled by the Reviewer | Goes to home page of the Reviewer | Pass |

***Table6.1*** Reviewer login

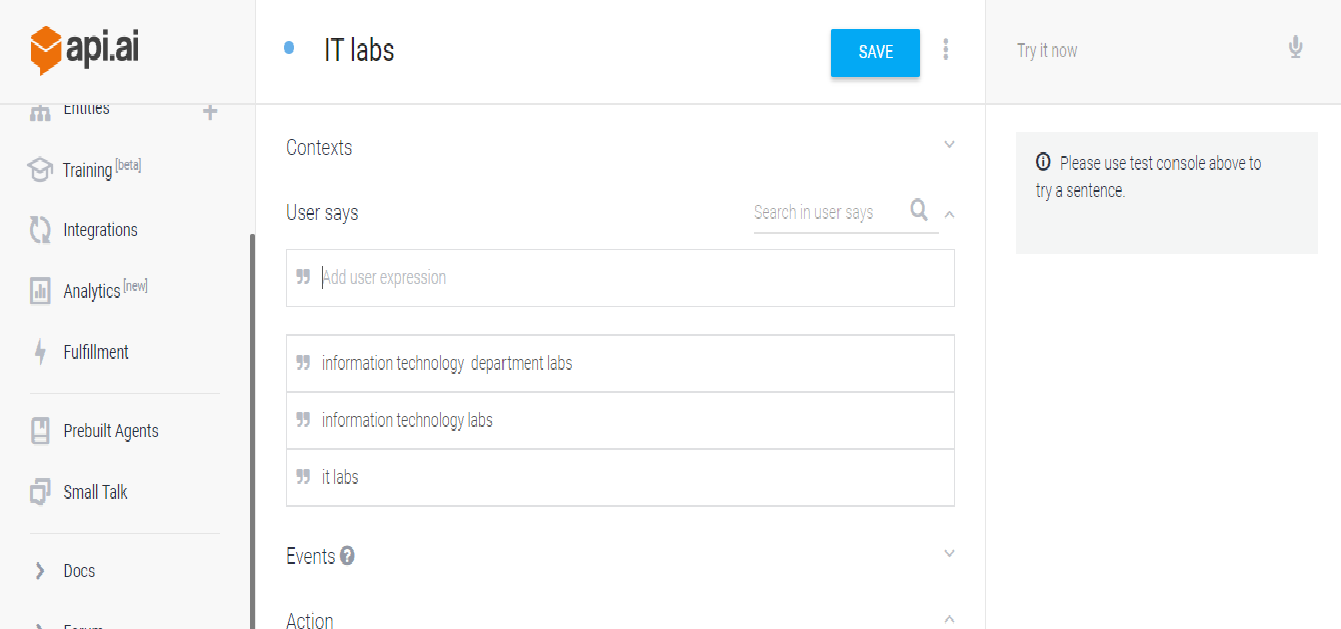
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TC ID** | **Input** | **Description** | **Expected Results** | **Pass/Fail** |
| TC01 | Sign-in  No connection | When user did not connect with the bot | Ask the related queries | Fail |
| TC02 | Sign-in  Not integrated messenger | When user sign-in the messenger, but it was not integrated. | Please check your bot | Fail |
| TC03 | Sign-in  Integrated messenger  Unrelated query | User sign-in in the integrated messenger but the query is unrelated | Please ask the related information | Fail |
| TC04 | Sign-in  Integrated messenger  Related query | Valid query | Respond for the query | Pass |

***Table6.2*** Users Bot

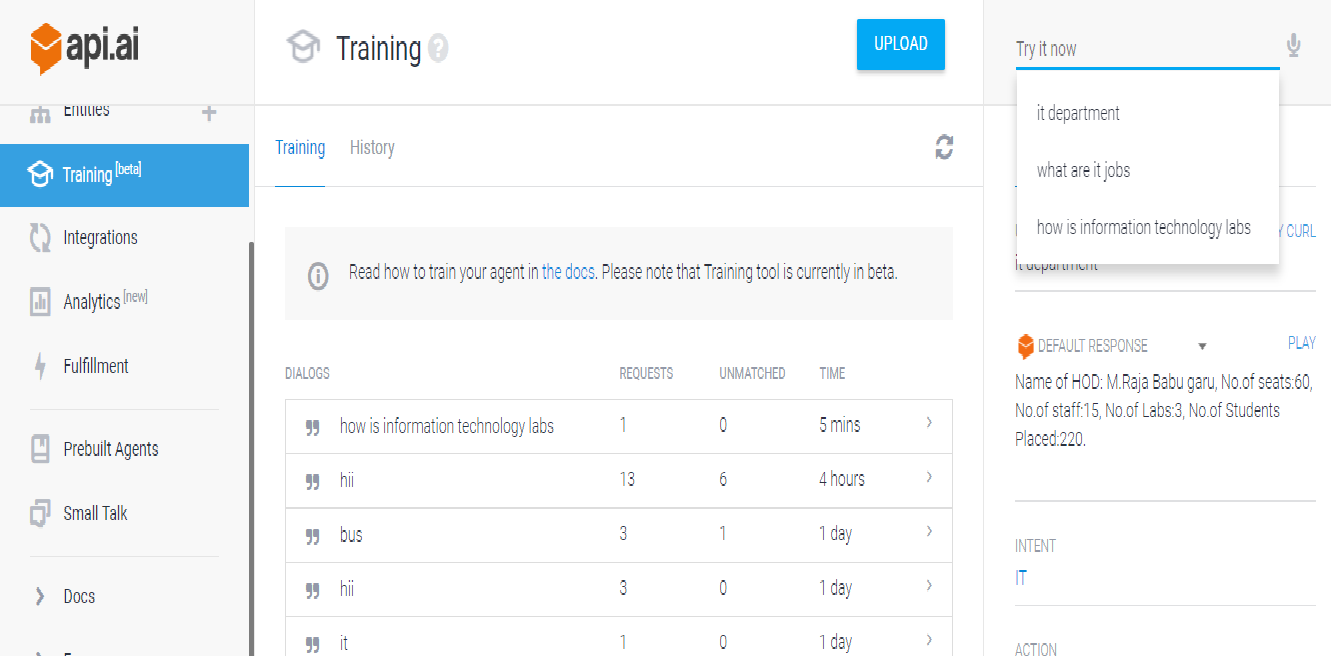
**7. SCREENSHOTS**

****

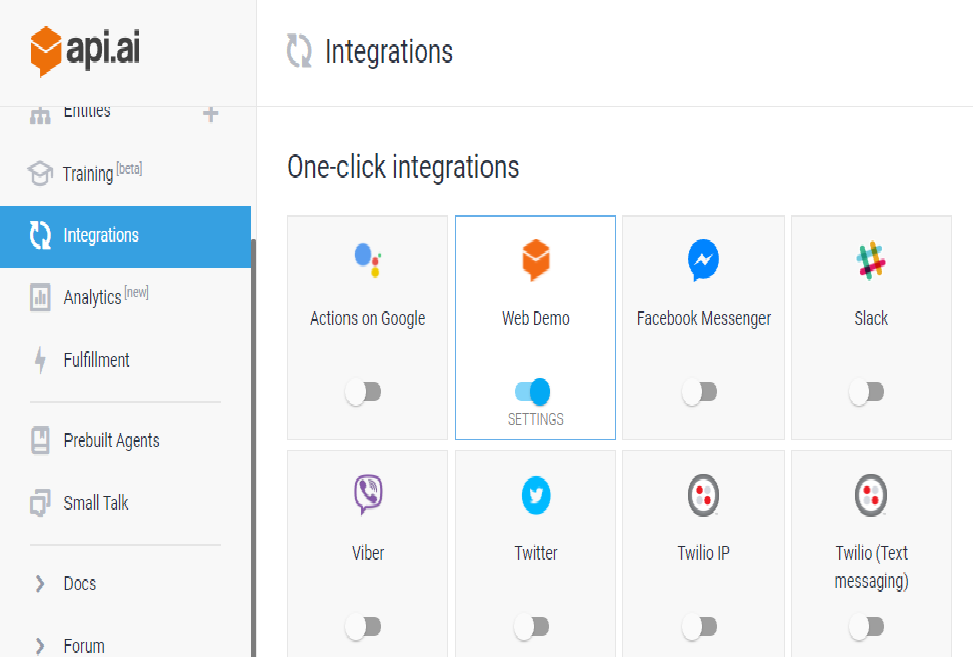
**Screen7.1**Creating Agent



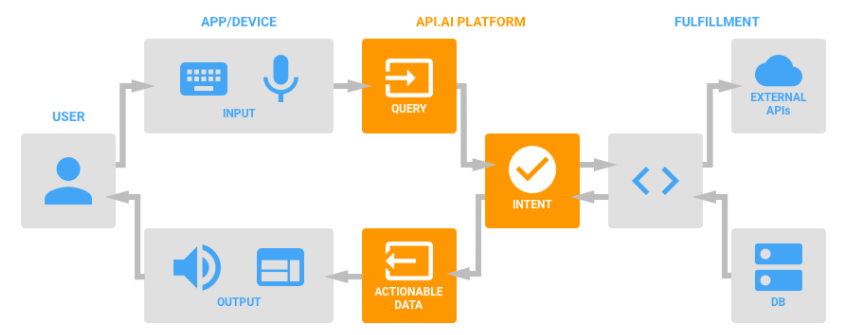
**Screen7.2**Creating Intents

****

**Screen7.3**Training Bot

****

**Screen7.4:** Integration

****

**Screen 7.5 Block Diagram**

**8. CONCLUSION & FUTURE SCOPE**

**8.1 CONCLUSION**

The main objective of this bot is to provide services to the users by giving the related information of our college in stipulated time.

**8.2 FUTURE SCOPE**

The Chatbot is to decrease the work burden and easier to publish our college strength and make it strong in technical way.

This service was introducing to save the time and reduce the process of enquiring about our college. The user (faculty/employee/student/others) can easily access this bot by connecting with the messenger and ask any related information about this college and can get quick response from the bot. While replaying the bot, the developer not need to be in online because it will give the response automatically based on users expressions which already trained by the developer.

**9. BIBLIOGRAPHY**

**References Made From:**

CLOUD SERVICES: - API

**Sites Referred:**

1. <https://api.ai/>
2. <https://api.ai/docs/getting-started/basics>
3. <https://api.ai/docs/integrations/>
4. <https://en.wikipedia.org/wiki/API.AI>