

Faculty of Technology, Department of Computer Engineering

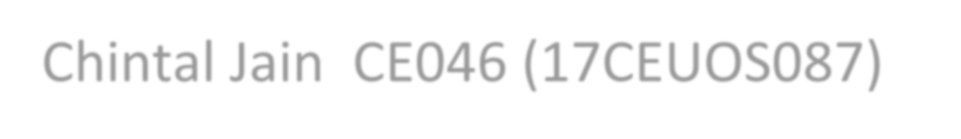
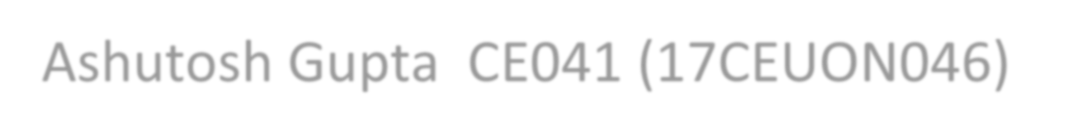
## B. Tech. CE Semester – V

Subject: (CE – 515) Advanced Technologies Project Title:

## NOTION

(The World Of Idea)

By:



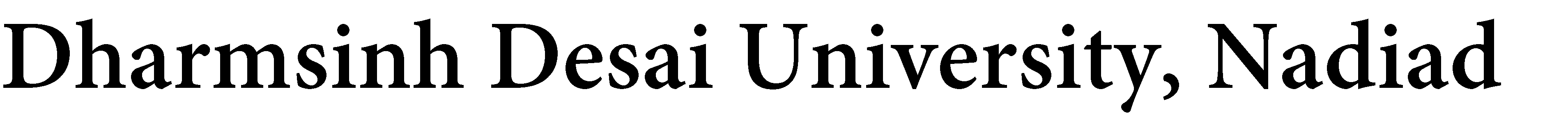
# Zalavadiya Nemish ( 17CEUOG074 )

# Vyas Vipul ( 17CEUBG009 )

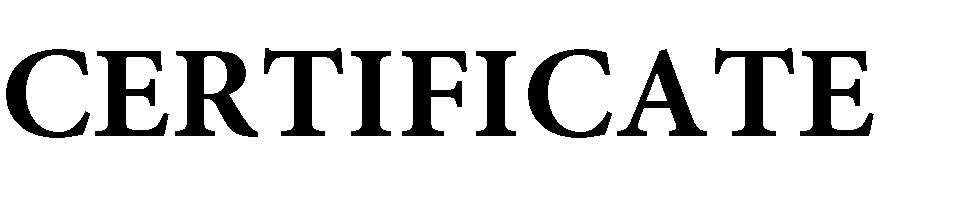


 Prof. Pinkal C. Chauhan





Faculty of Technology, Department of Computer Engineering



This is to certify that Advanced Technologies Project entitled “**Notion**” is the bonafied report of work carried out by

**1) Nemish Zalavadiya (17CEUOG074)**

**2) Vipul Vyas (17CEUBG009)**

Of Department of Computer Engineering ,Semester V , academic year 2019-2020, under our supervision and guidance.



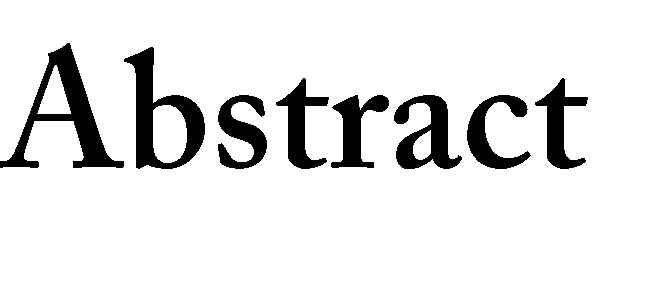
|  |  |  |
| --- | --- | --- |
| **Prof. Pinkal c. Chauhan**  Assistant Professor of Department of Computer Engineering, Dharmsinh  Desai University, Nadiad | **Prof. Parth R. Dave**  Assistant Professor of Department of Computer Engineering, Dharmsinh  Desai University, Nadiad | **Dr. C. K. Bhensdadia**  Head of Department of Computer Engineering, Dharmsinh Desai  University, Nadiad |

*Table of Contents*

1. Abstract 4
2. Introduction 5
   1. Project Details: Brief Introduction 5
   2. Technology and Tools Used 5
3. Software Requirement Specifications 6
   1. Scope 6
   2. System Functional Requirements 6
   3. Other Non‐Functional Requirements 13
4. Design 14
5. Implementation Details 31
   1. Modules 31
   2. Functional prototypes 32
6. Testing 34
   1. Testing Method 34

6.4 Test Cases 34

1. Screen‐shots of the System 36
2. Conclusion 39
3. Limitations of System 40
4. Future Extensions to the System 40
5. Bibliography 41



**Notion** is a [Idea](https://en.wikipedia.org/wiki/Note-taking) sharing and [collaboration](https://en.wikipedia.org/wiki/Collaborative_software) application with export support that also integrates tasks, wikis, and databases.

**Notion** is an online publishing platform. The platform is an example of social journalism, having a hybrid collection of amateur and professional people and publications, or exclusive blogs or publishers on Notion and is regularly regarded as a blog host.

Notion is an idea to allow and provide platform to the people

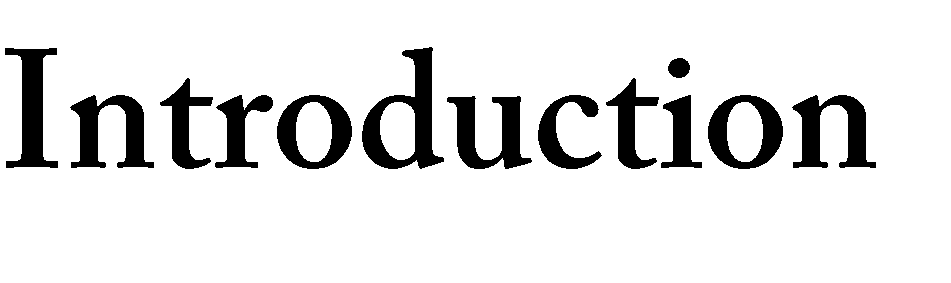
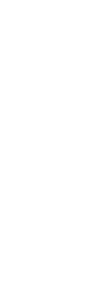
To share their ideas throw our application and get best response by other interested people. Here we are providing easy registration and login system for the people to upload their ideas to the other users and

Read there Articles with no cost.

**Notion**

The all-in-one workspace for your notes, tasks and article sharing.

© 2019 Notion Labs, Inc.

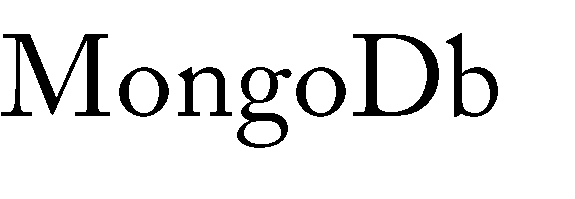
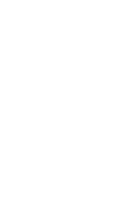
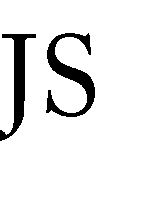
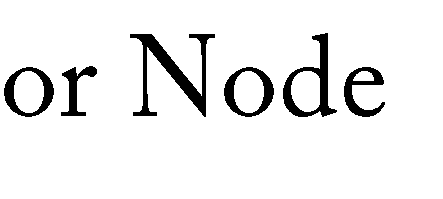
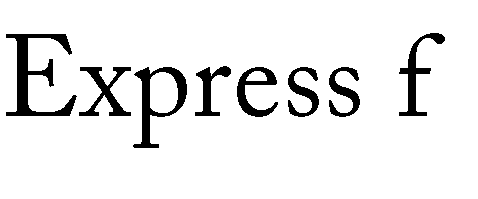
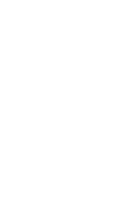
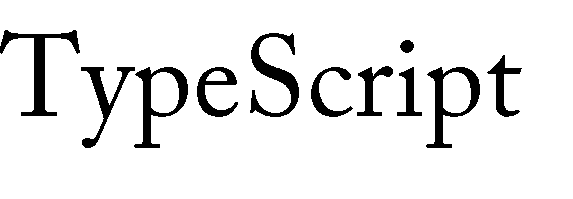
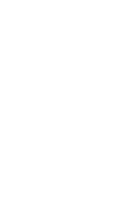
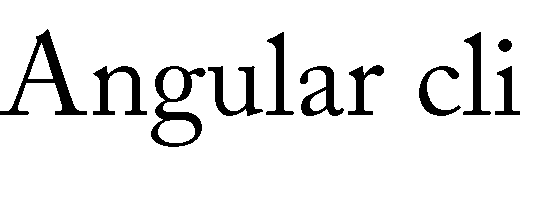
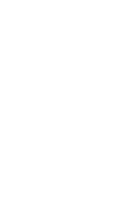
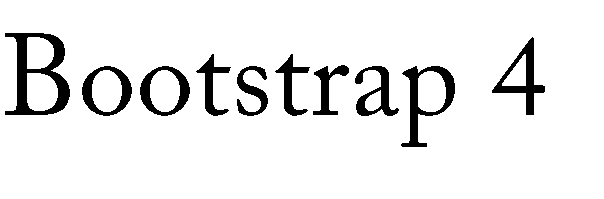
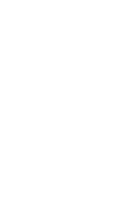
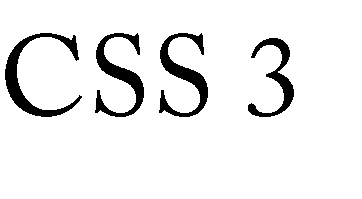
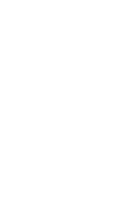
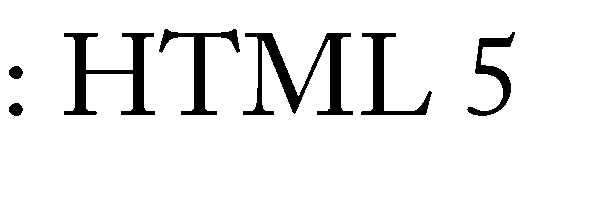
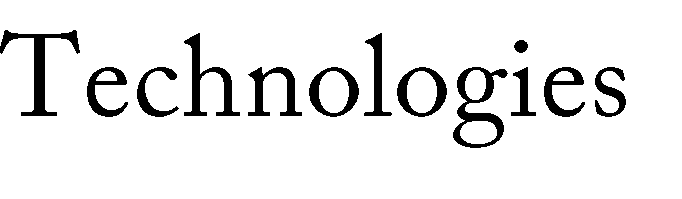


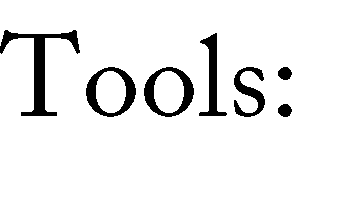
# Brief Introduction

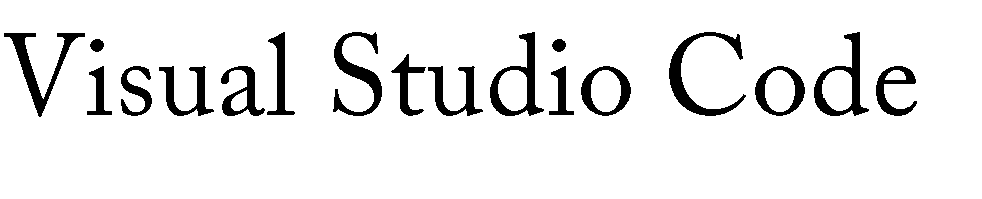
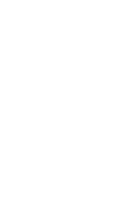


**Notion** is all about sharing a good quality knowledge from people to people to with experts support Who is already contributing to the people.

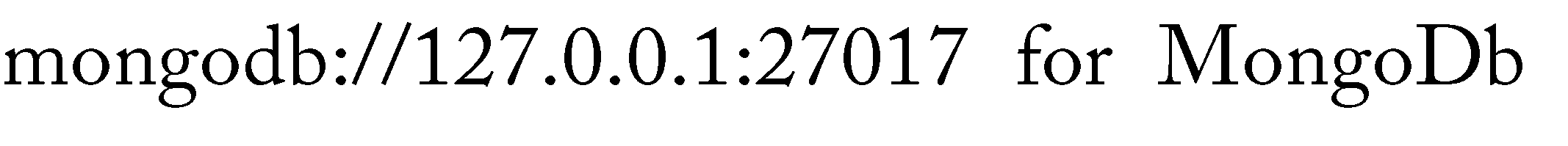
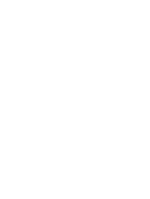
* 1. **Tools/Technologies Used**



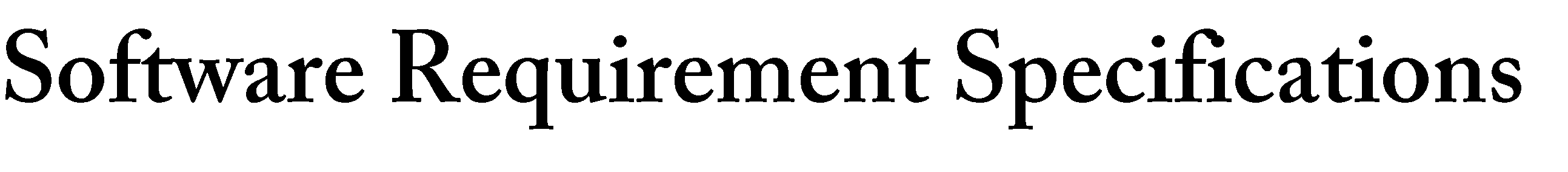
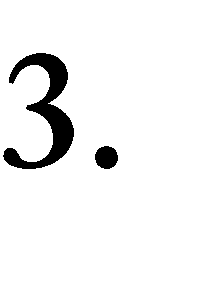




## 



localhost/3000 for nodejs

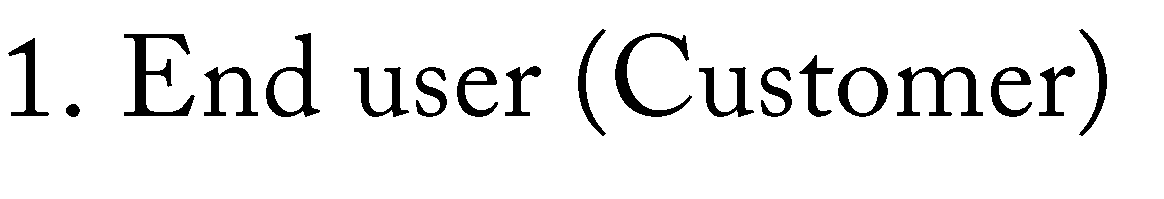


# Application Scope

Our system allow people to upload their ideas as a post so that other people can read it and get knowledge in particular field,

And also can improve their knowledge quality.

* 1. **Types of Users:-**



2. Admin

**System Functional R equirements**

R1. User Authentication

Description : In this Module,User can Register for new Account or login with right Credentials.If user login successfully then Auth\_Page will Redirect to Home page.

R1.1 : Sign in

INPUT : Credentials

OUTPUT : DashBoard / Message

PROCESS : Verifying Credentials

R1.2 : Sign up

INPUT : Credentials

OUTPUT : DashBoard / Message

PROCESS : Data [ Credentials ] insertion in database

R2. User Profile Management

Description : In this Module,User can See personal information ,change password ,update personal information and User can Upload his/her own Articles.In Article, user can upload Text ,Image ,Video[Optional].User can update the uploaded Article ,Delete the Article ,Show the uploaded Articles.

R2.1 : Select for any information.

INPUT : User selection

OUTPUT : Information page

PROCESS : Retrieving Particular Personal information

R2.2 : Change or Update the Settings

R2.2.1 : Change Password

INPUT : Current and New Password

OUTPUT : Message

PROCESS : Verifing Credentials and Updating

R1.2.2 : Update Personal Information

INPUT : New data of Personal Information

OUTPUT : Message

PROCESS : Data of Personal Information insertion in database

R2.3 : Manage MyArticals

R2.3.1 : Create

INPUT : Article and Image related to it [ video optional ]

OUTPUT : Message

PROCESS : Uploading Data

R2.3.2 : Update

INPUT : Changes in Article

OUTPUT : Message

PROCESS : Data Updation in database

R2.3.3 : Delete

INPUT : User Selection

OUTPUT : Message

PROCESS : Data Deletion in database

R2.3.4 : Show

INPUT : Artical Information

OUTPUT : Artical

PROCESS : Data Retrieves from database And Show

R3. Searching Articles

R3.1 : Categories

INPUT : Keyword

OUTPUT : DashBoard With recommended Articles for Type

PROCESS : Searching In database with keyword as Category.

R1.2 : Username

INPUT : Keyword as Username

OUTPUT : DashBoard With Articles of Provided Username

PROCESS : Searching In database with keyword as User

Description : In this Module,User can Search different Articles by Categories or Username/Author\_name.

R4. DashBoard Home

R4.1 : Article

INPUT : AFter Authenticating This page is Loaded

Automatically

OUTPUT : DashBoard With Articles by Provided Interest

PROCESS : Searching In database with Interest Parameter

R4.2 : Categories

INPUT : User Selection

OUTPUT : DashBoard With Articles Of User Selection

PROCESS : Searching In database with Interest Parameter

Description : In this Module ,User can See his/her Articles with Category of interest and Go To Profile and Search for Particular

Area of Categories.

R5. Follow Authors/Users

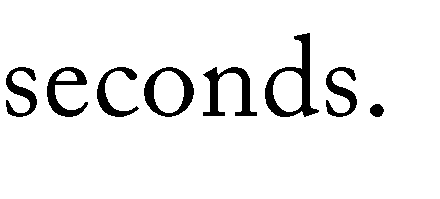
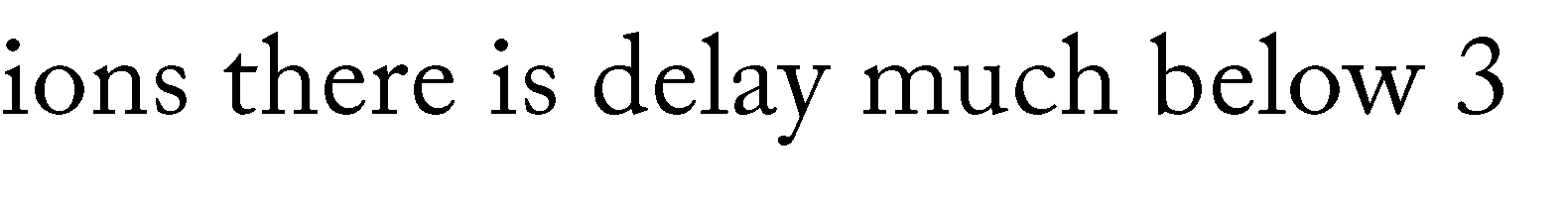
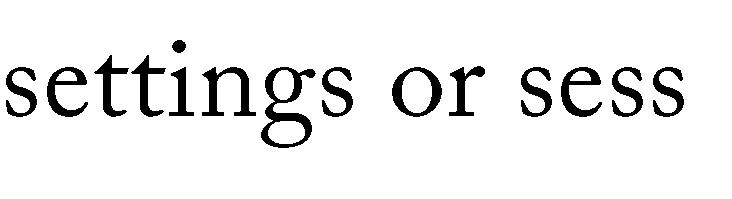
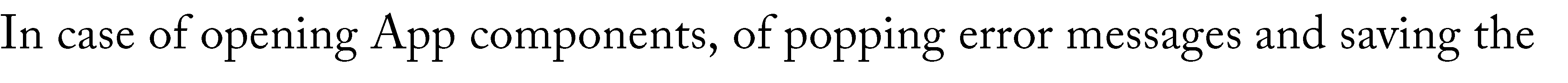
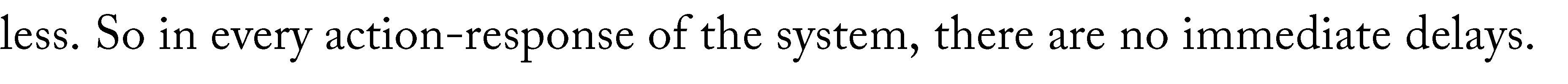
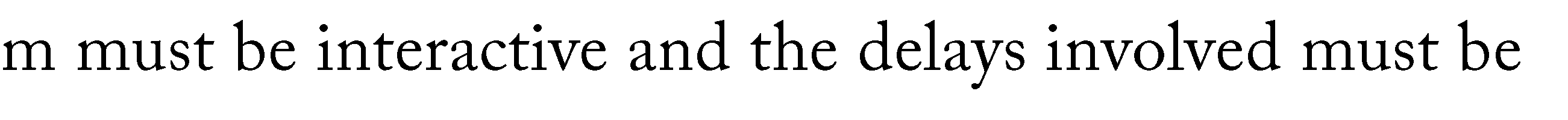
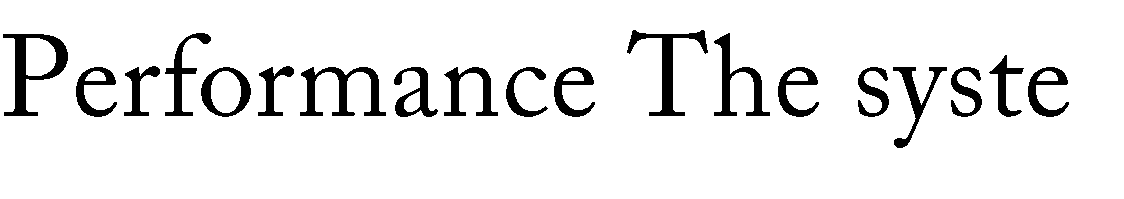
INPUT : User Selection

OUTPUT : Message

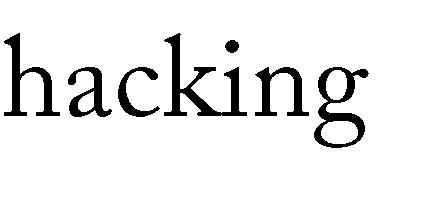
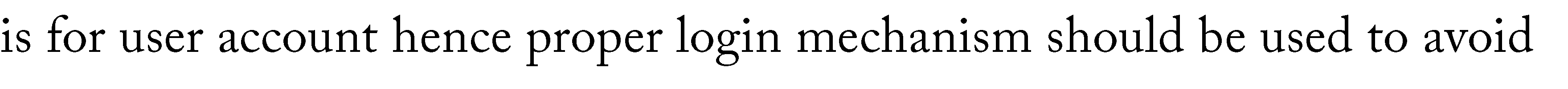
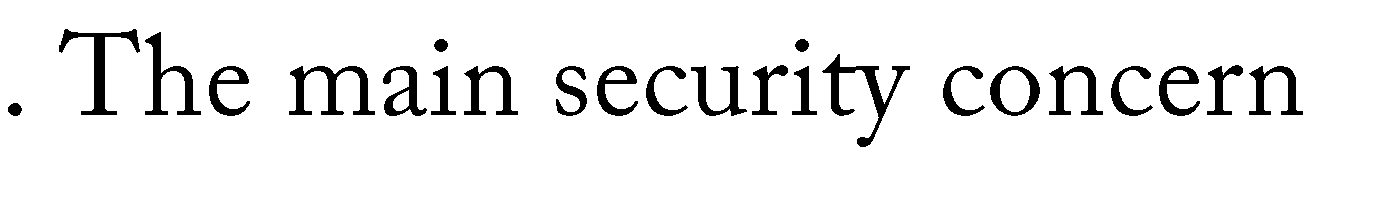
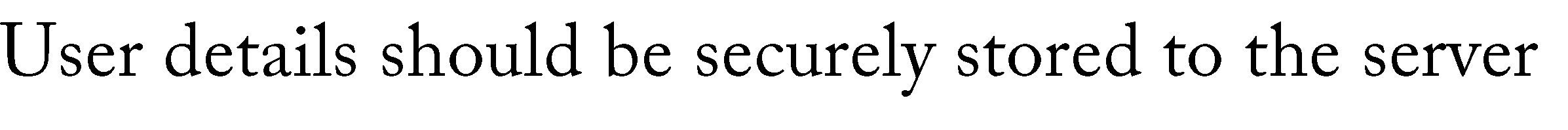
PROCESS : Add follow information to database for Recommendation.

* 1. **Other Nonfunctional Requirements**

1. **Performance**

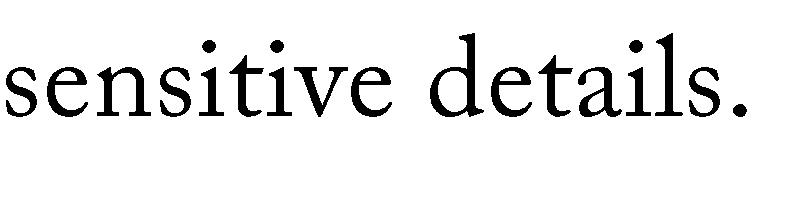
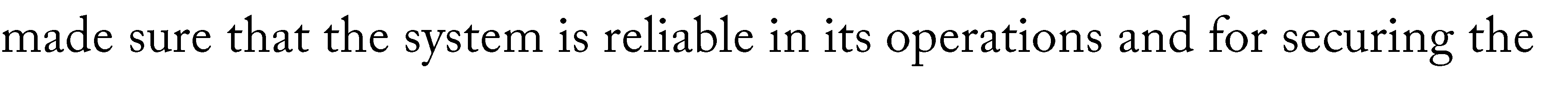
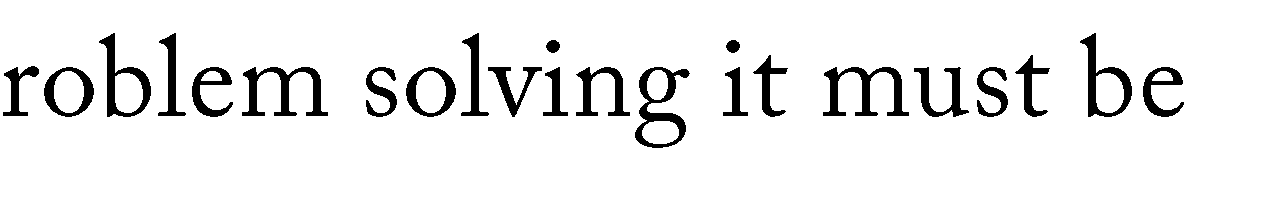
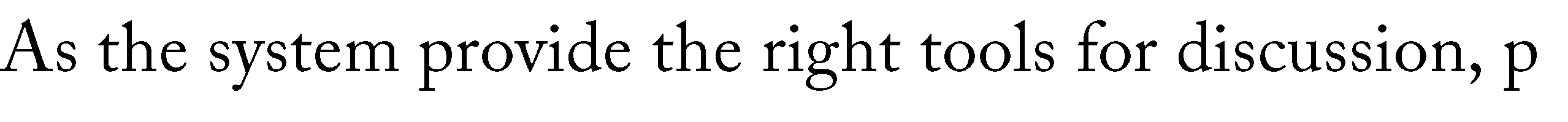


1. **Safety**

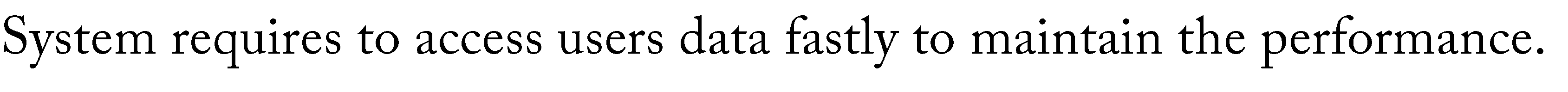


.

1. **Reliability**



1. **Database**



# Data Dictionary

Categories

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table |  |
| 1. | C\_ID | Number | 20 | YES | YES | PK |  |  |
| 2. | C\_Name | Varchar2 | 20 | YES | YES |  |  |  |

Articles

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table |  |
| 1. | AI\_ID | Number | 20 | YES | YES | PK |  |  |
| 2. | ARTICLE\_PATH | Varchar2 | 50 | YES |  |  |  |  |
| 3. | IMAGE\_PATH | Varchar2 | 50 | YES |  |  |  |  |
| 4. | DATE | Date |  | YES |  |  |  |  |

Users

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table |  |
| 1. | Username | Varchar2 | 50 | YES | YES | PK |  |  |
| 2. | Password | Varchar2 | 50 | YES |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3. | type | Varchar2 | 10 | YES |  |  |  |  |

Profile

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table |  |
| 1. | Username | Varchar2 | 50 | YES | YES | FK | Users |  |
| 2. | Fname | Varchar2 | 10 | YES |  |  |  |  |
| 3. | Mname | Varchar2 | 10 | YES |  |  |  |  |
| 4. | Lname | Varchar2 | 10 | YES |  |  |  |  |
| 5. | Email\_id | Email | 40 | YES | YES |  |  |  |
| 6. | Interest\_Field | Varchar2 | 10000 | YES |  |  |  |  |
| 7. | Follow | Varchar2 | 10000 |  |  |  |  |  |

Upload\_Details

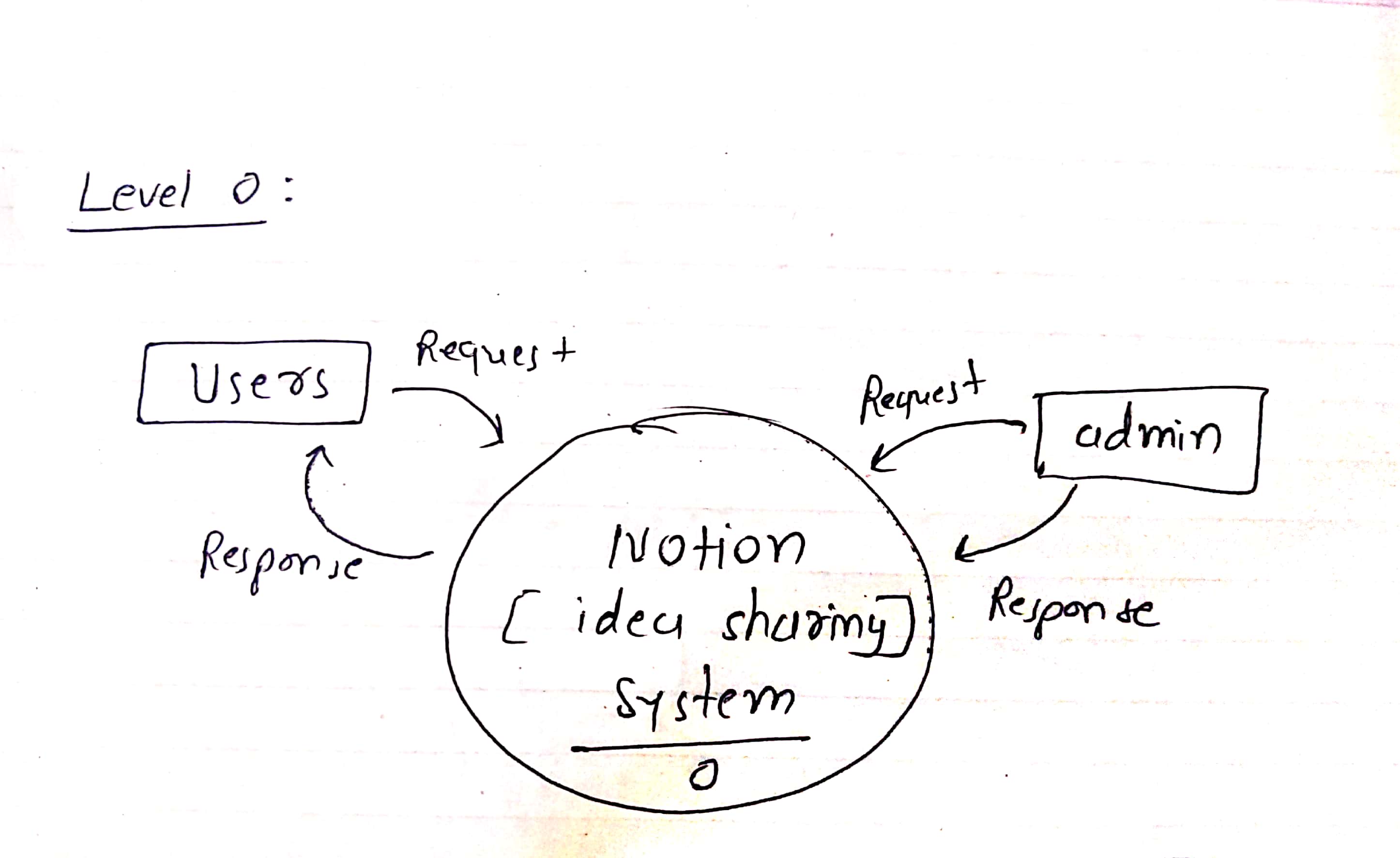
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table |  |
| 1. | AI\_ID | Number | 20 | YES | YES | FK | Articles |  |
| 2. | C\_ID | Number | 20 | YES |  | FK | Categories |  |
| 3. | Username | Varhcar2 | 50 | YES |  | FK | Users |  |

Video\_Details

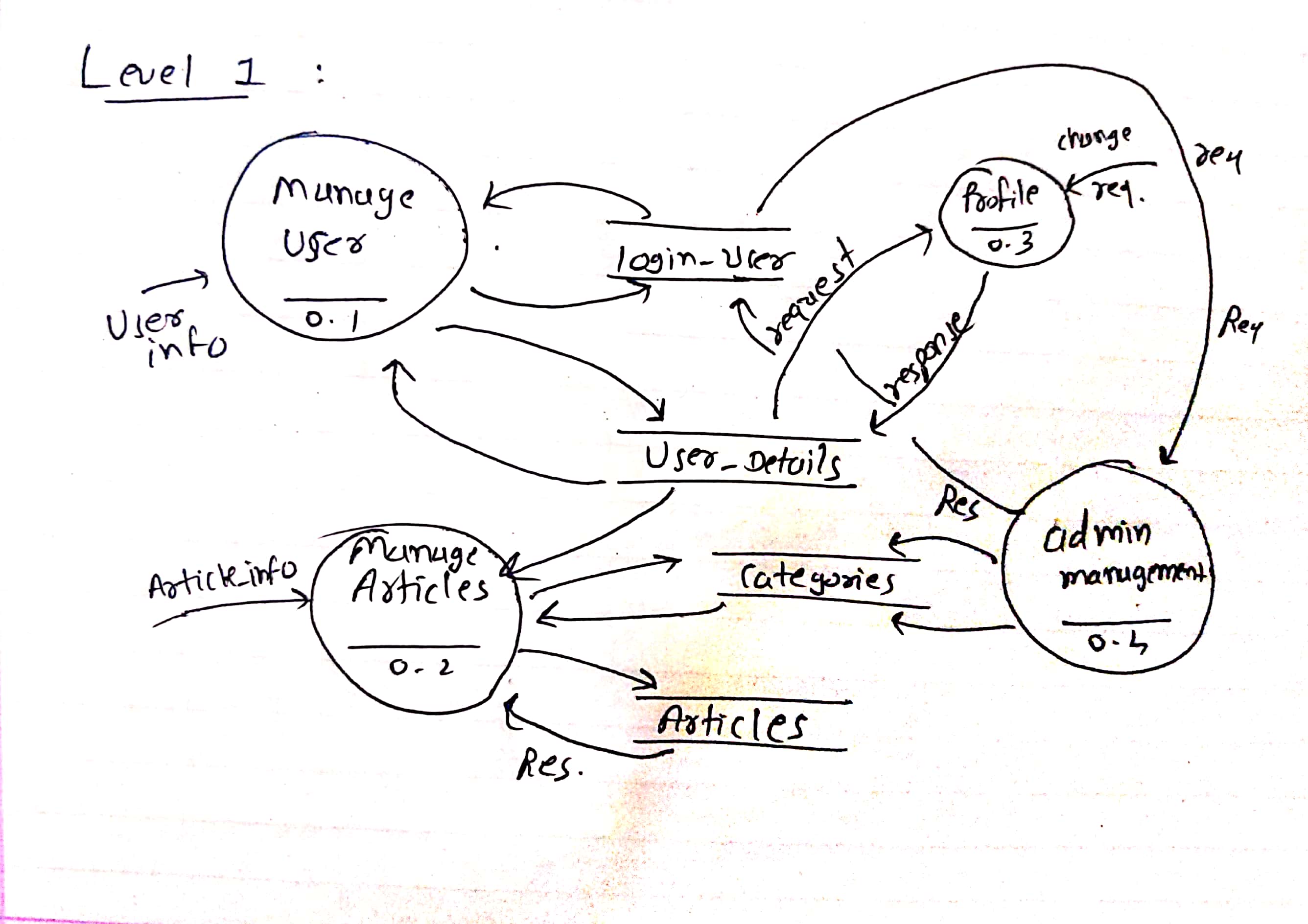
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK | Referenced Table |  |
| 1. | AI\_ID | Number | 20 | YES | YES | FK | Articles |  |
| 2. | Video\_Path | Varchar2 | 50 |  |  |  |  |  |
| 3. | Date | Date |  | YES |  |  |  |  |

# DFD

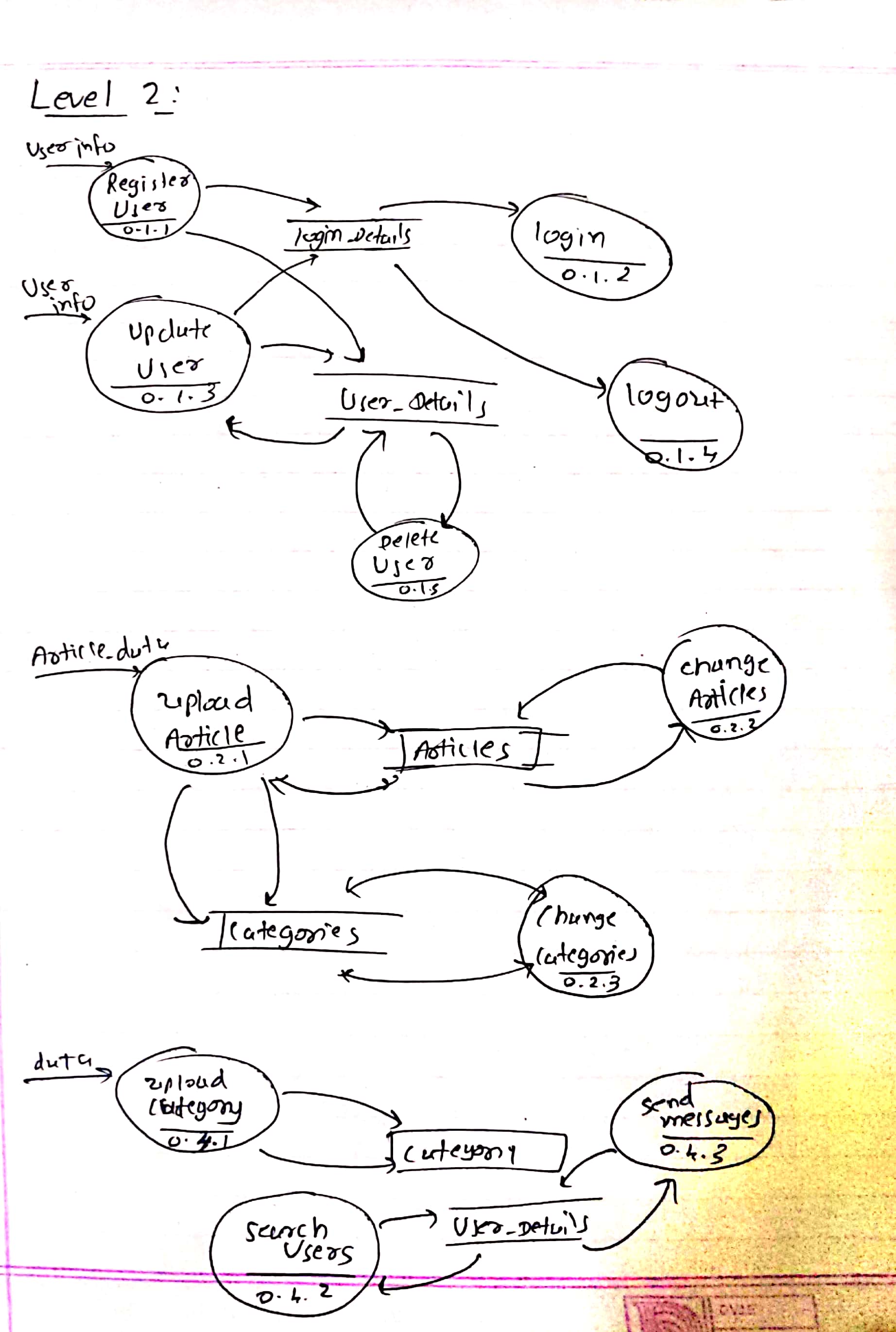
Level 0:



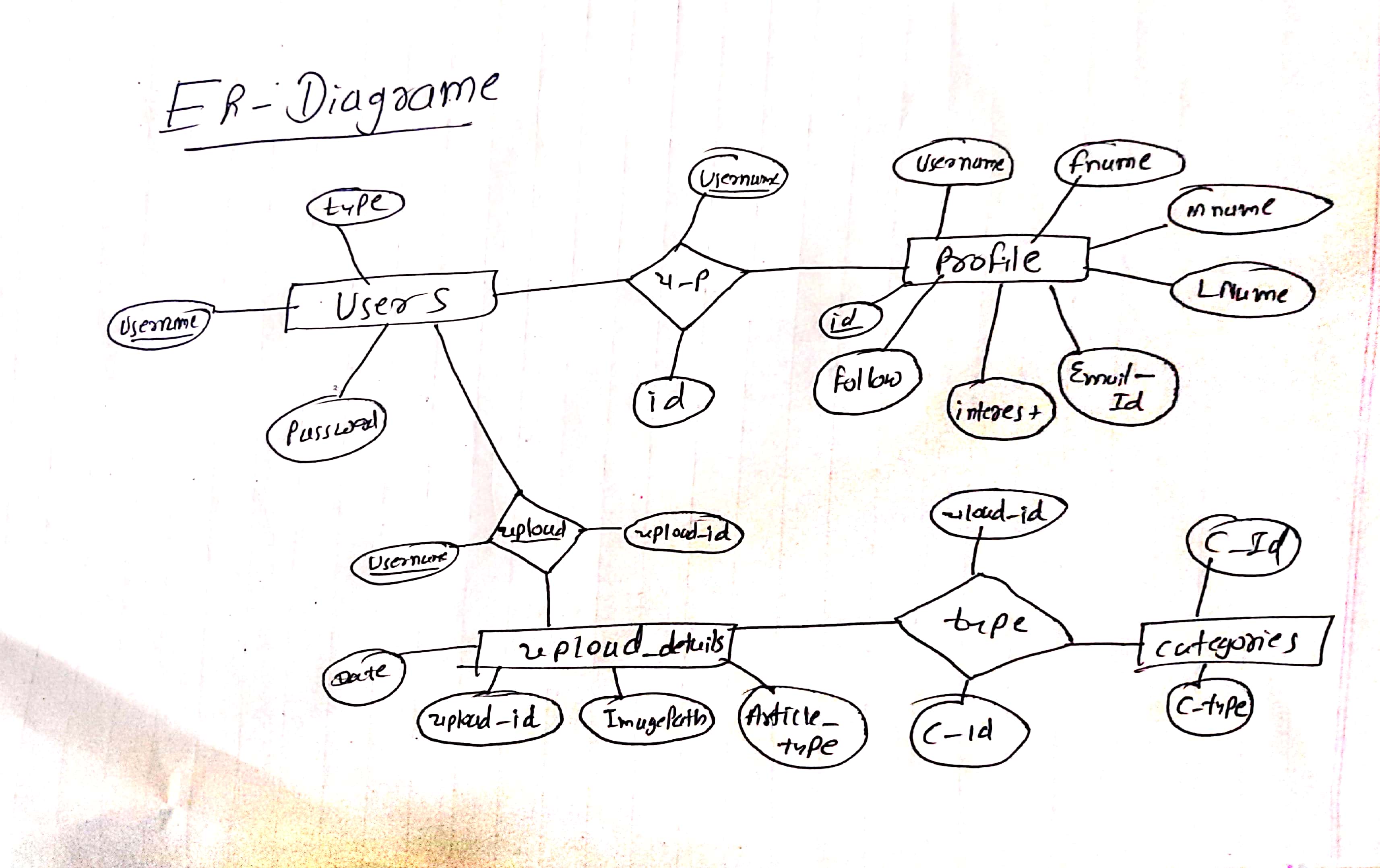
Level 1:

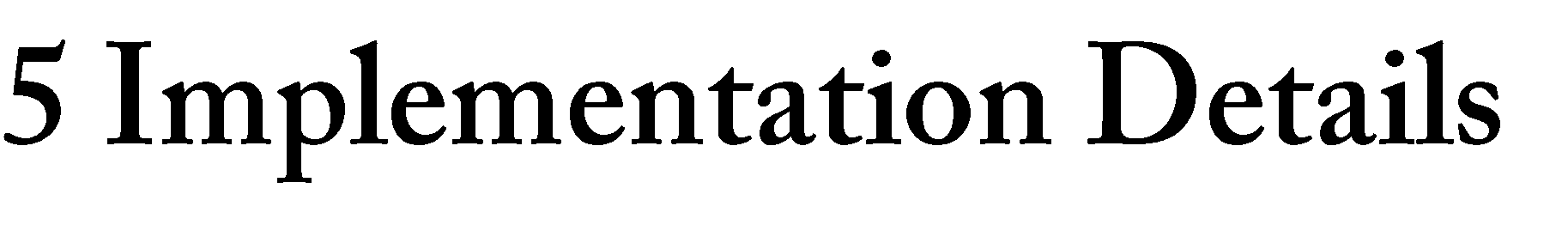


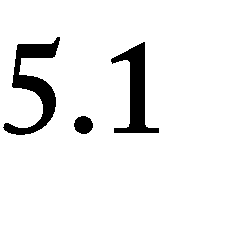
Level 2:

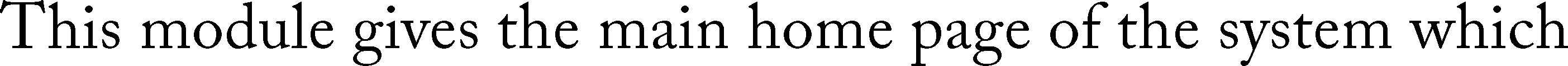
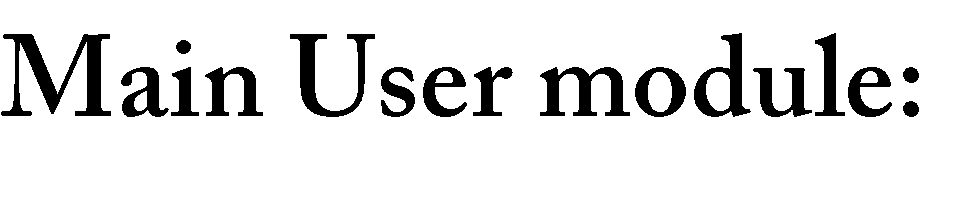


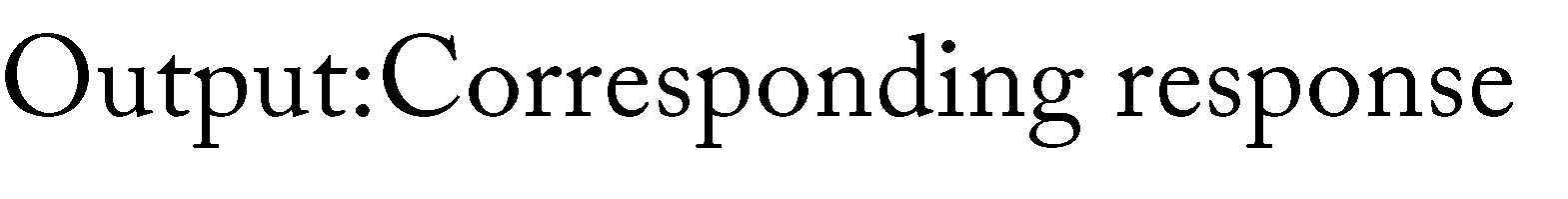
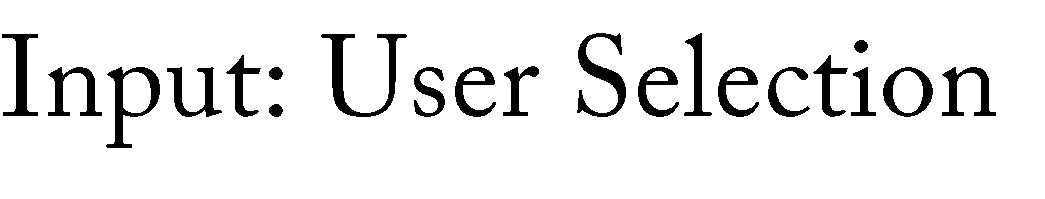
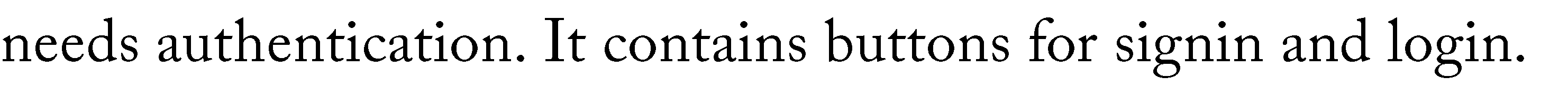
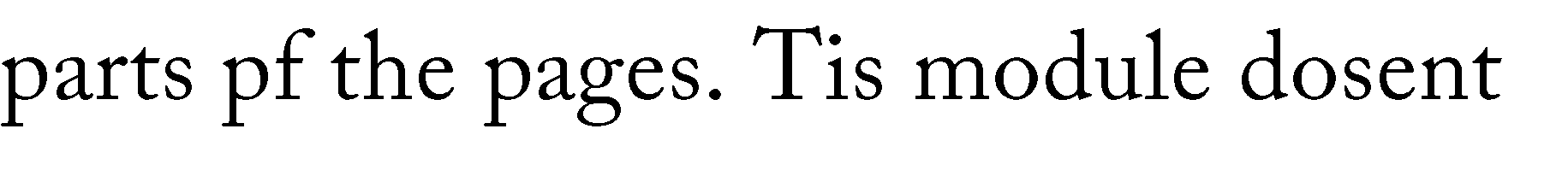
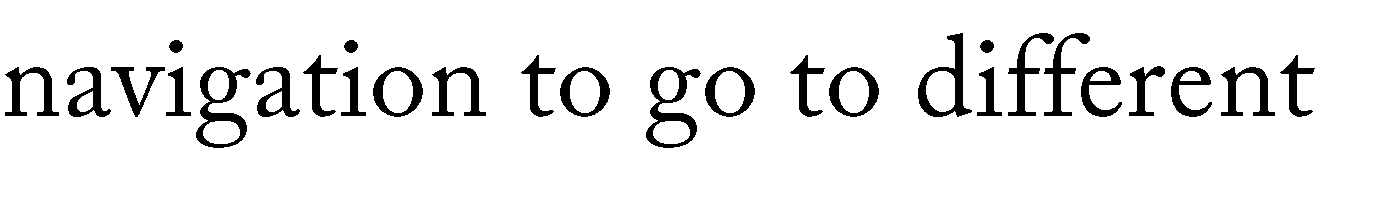
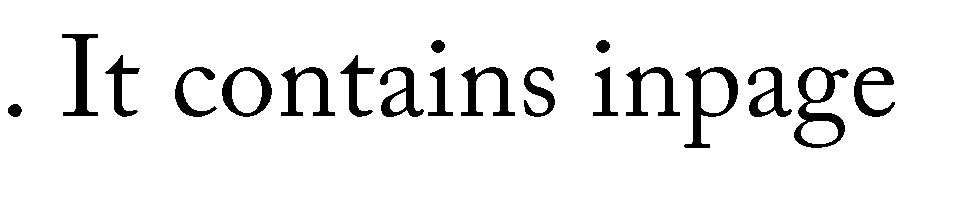
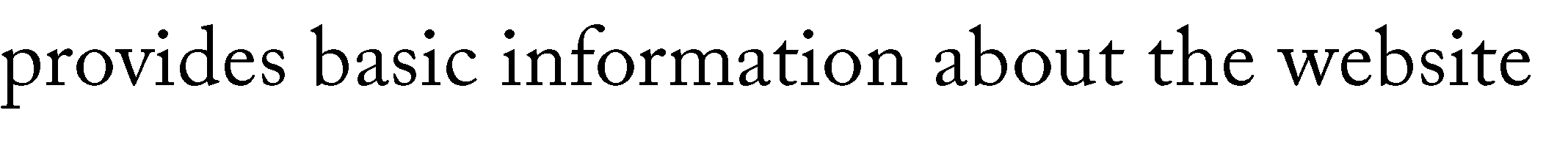
* 1. **E-R Diagram**

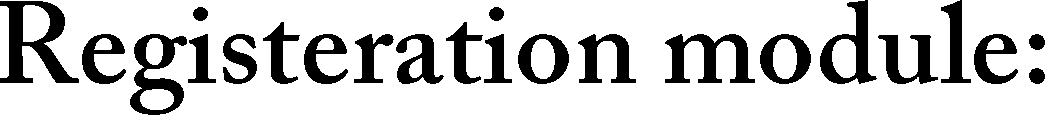


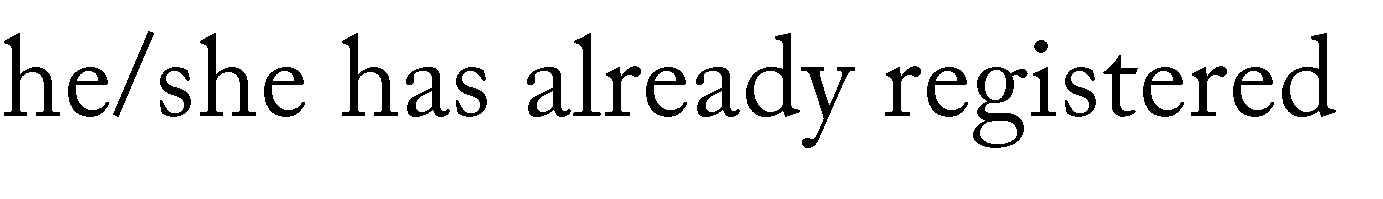
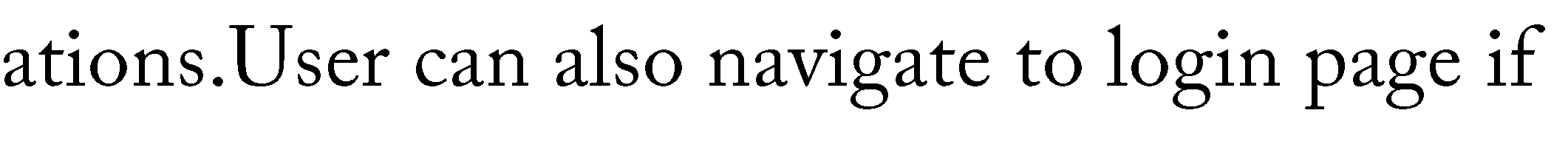
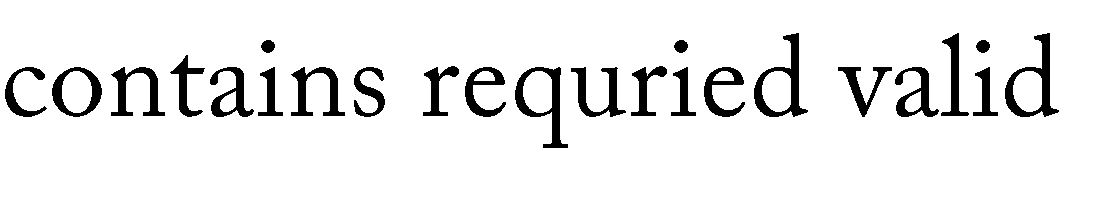
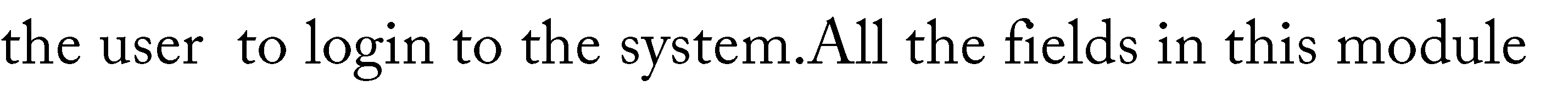
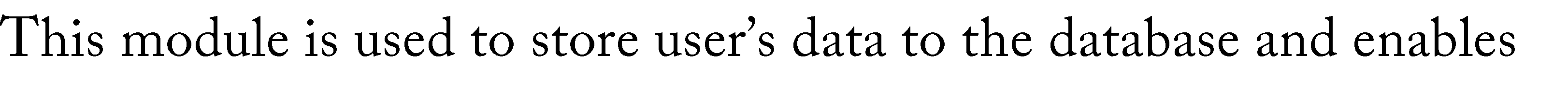


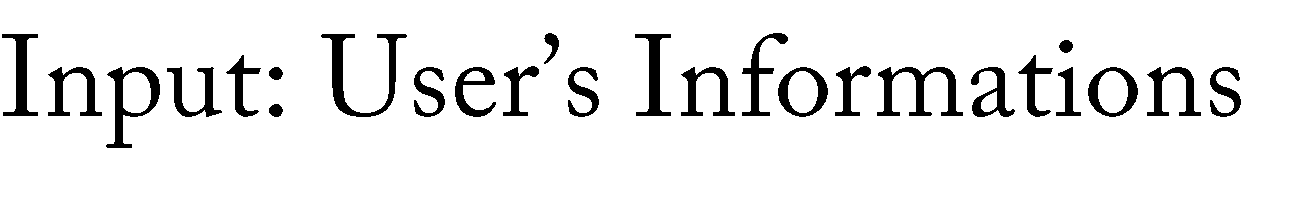
 **Modules Description**

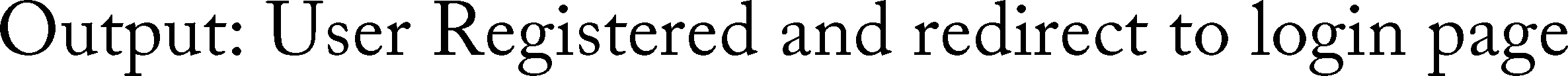
 

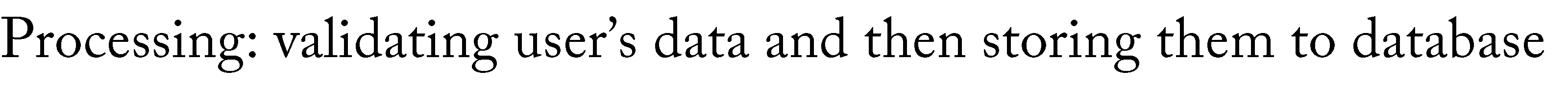


 

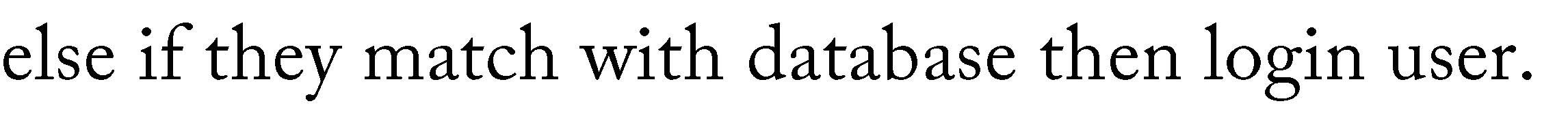
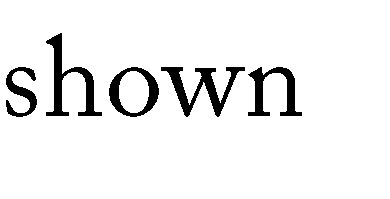
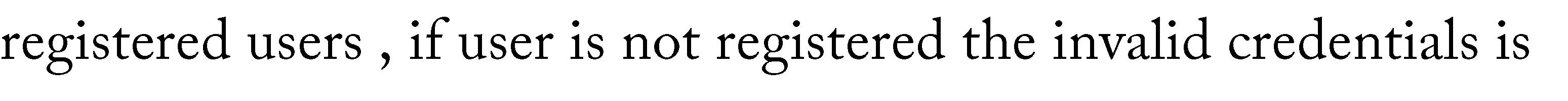
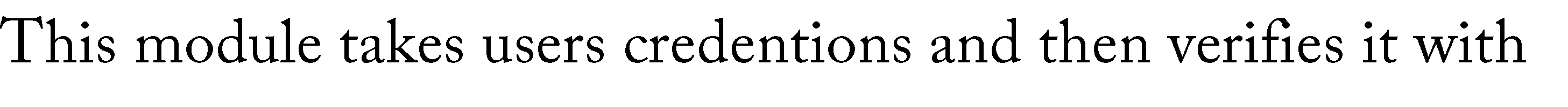
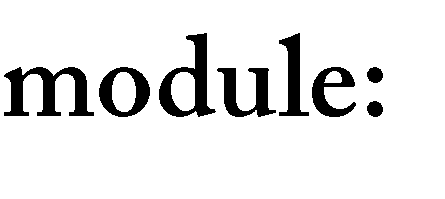
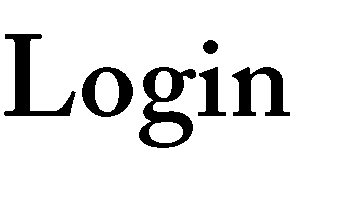


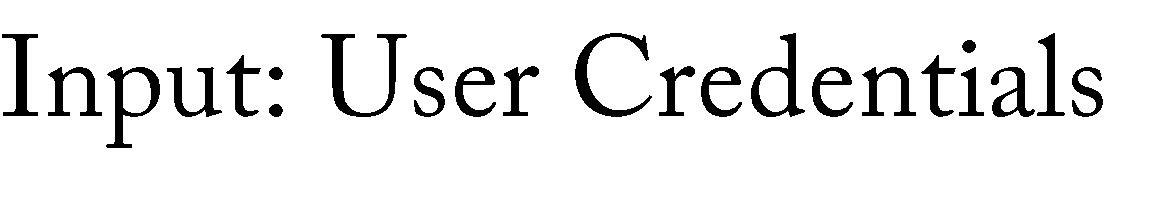


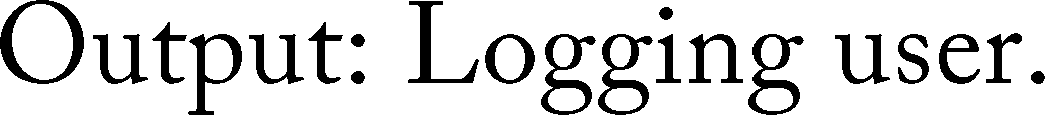


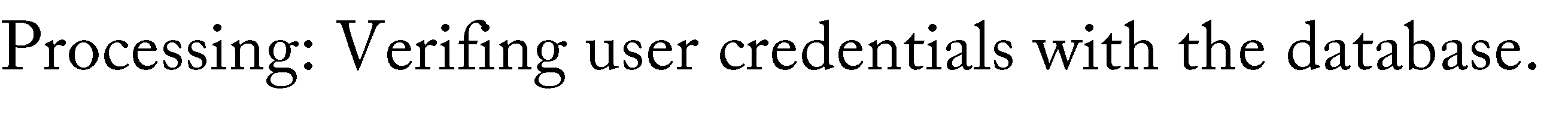




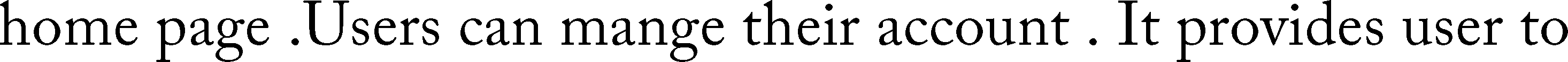


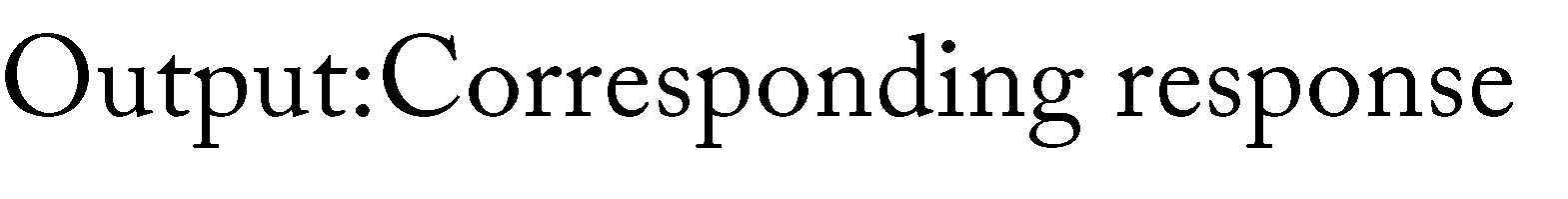
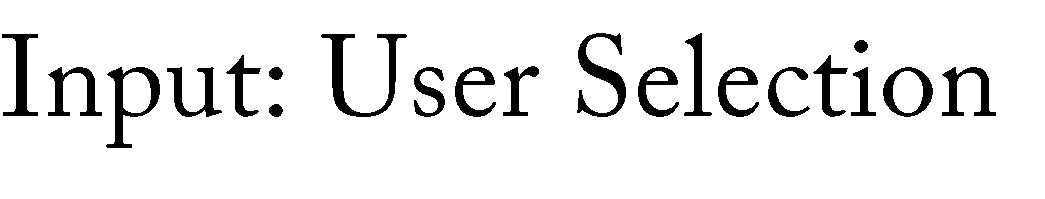
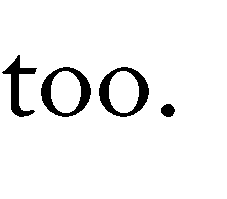
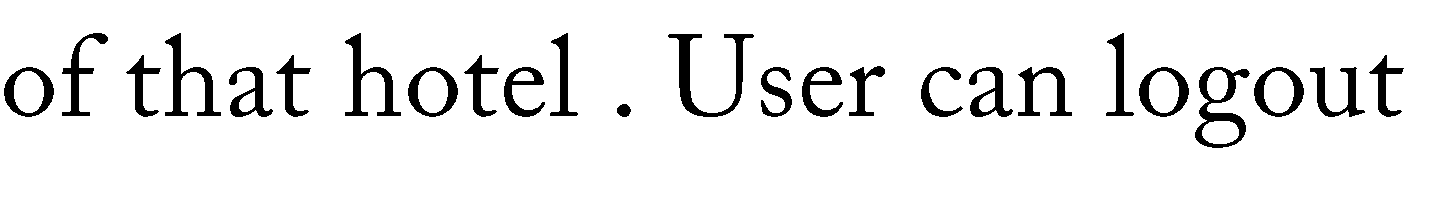
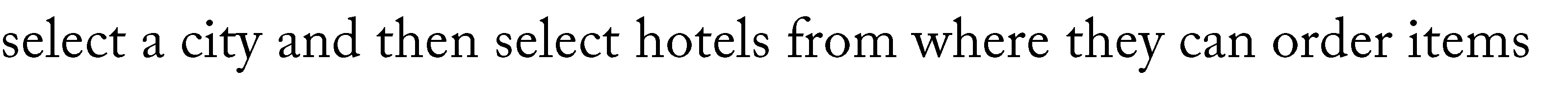


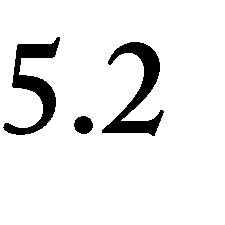


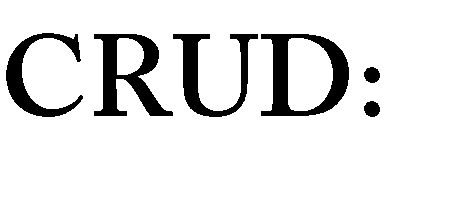


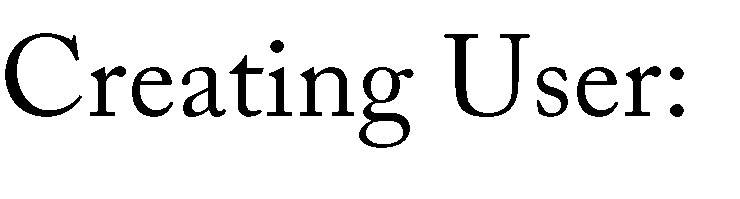
Content Module

This module is only can Access by the Authenticated users which is a



 **Function Prototypes**

 



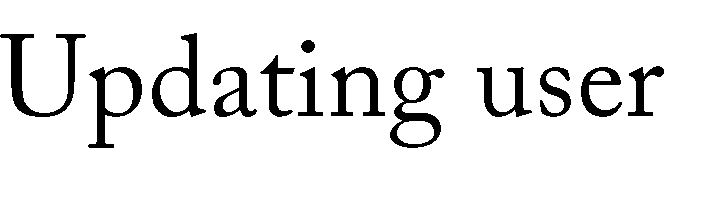
this.dbServ.registerUser(this.userData).subscribe(data=>{…}

registerUser(userData:UserDetails):Observable<any>

app.post("/user/create", function(req, res){

db.collection(‘users’).insertOne({username:req.body.user

,password:req.body.password},function(err){..}



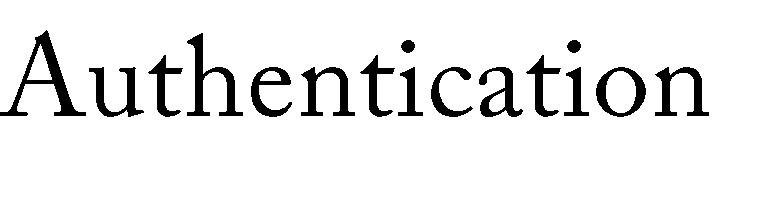
this.dbServ.UpdateUser(this.userData).subscribe(data=>{..}

UpdateUser(userData:UserDetails):Observable<any>

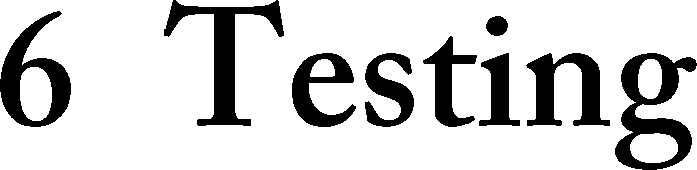
db.collection('User\_Details').update({email:req.body.email},{$set:{password:r eq.body.password}})

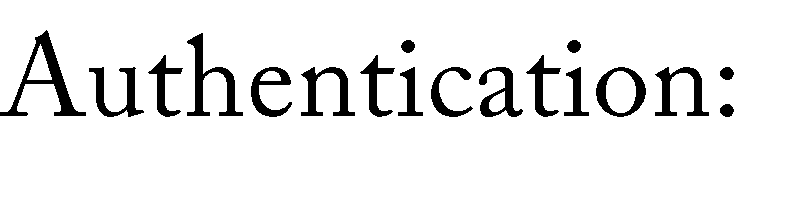
db.collection('Users).deleteOne({username:req.body.user},function(err){…

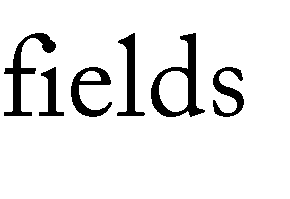
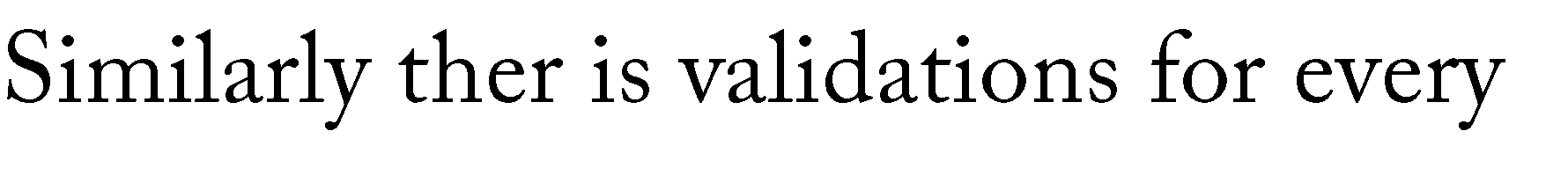
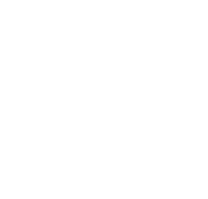
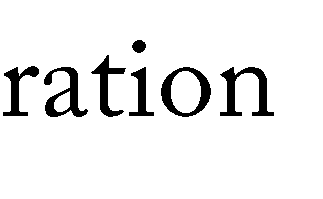
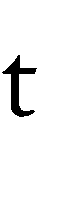
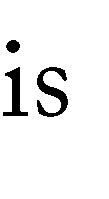
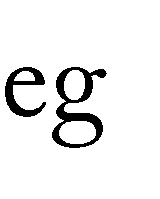
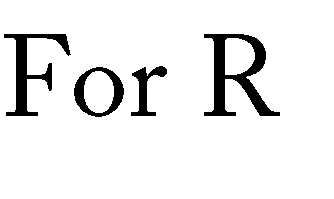
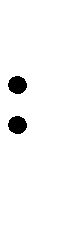
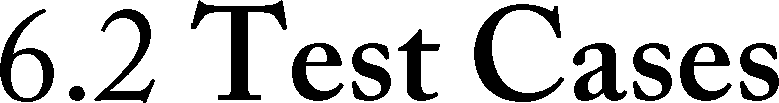
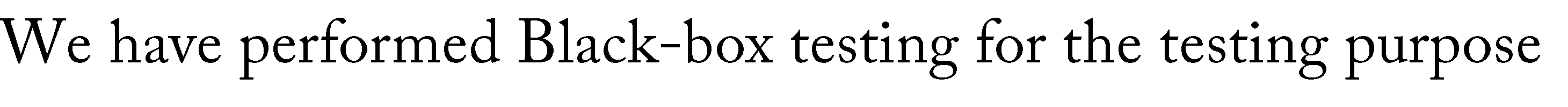
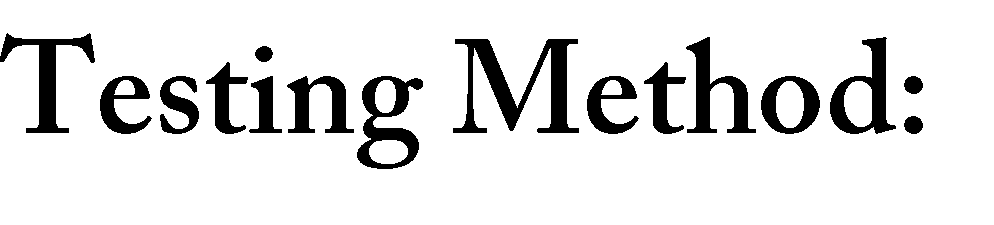
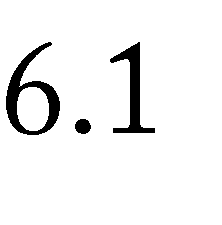
}

 

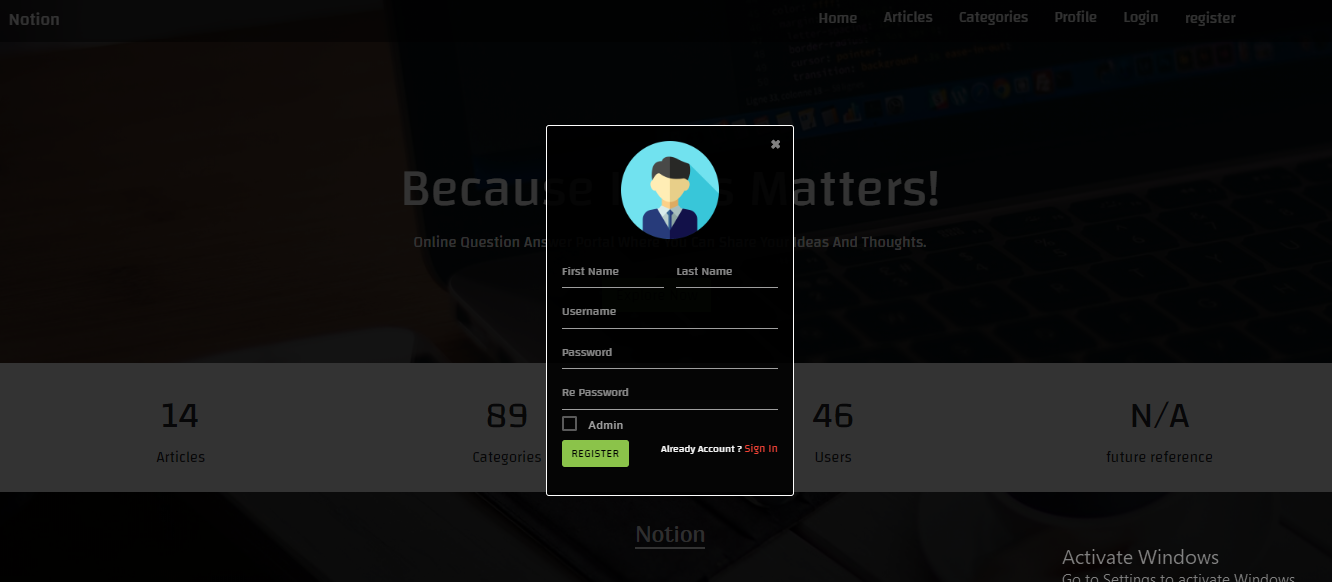
{path:'login',component: LoginComponent,canActivate:[Guard]},

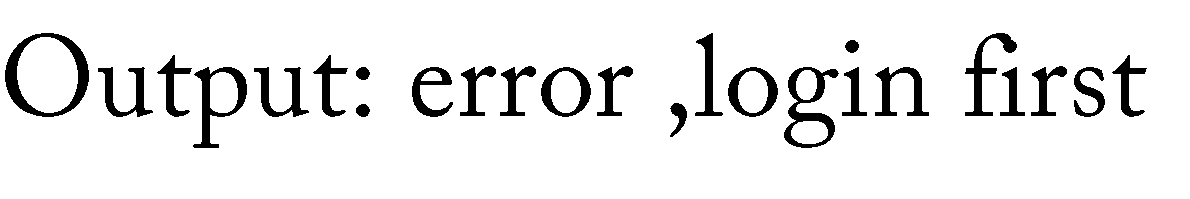
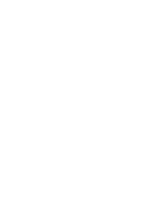
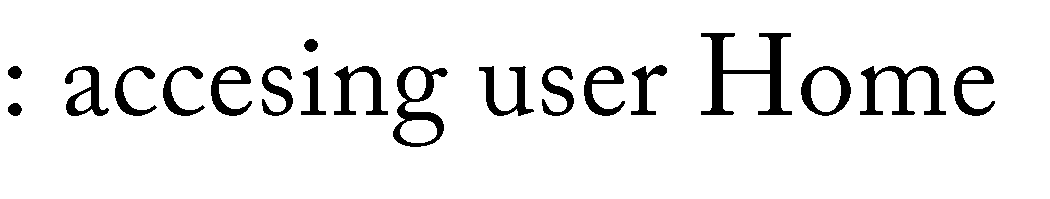
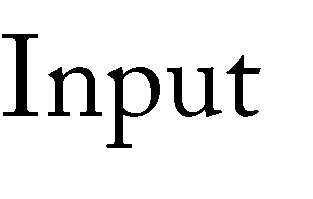
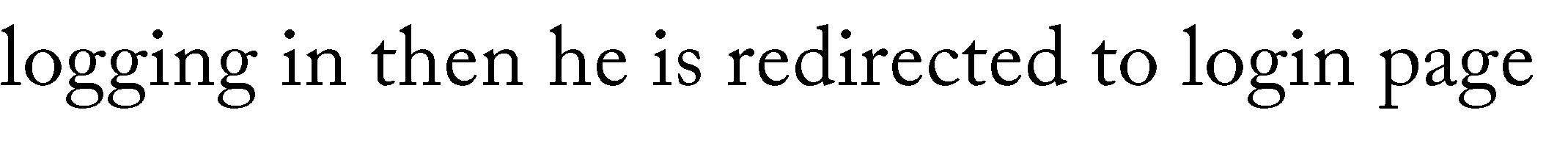
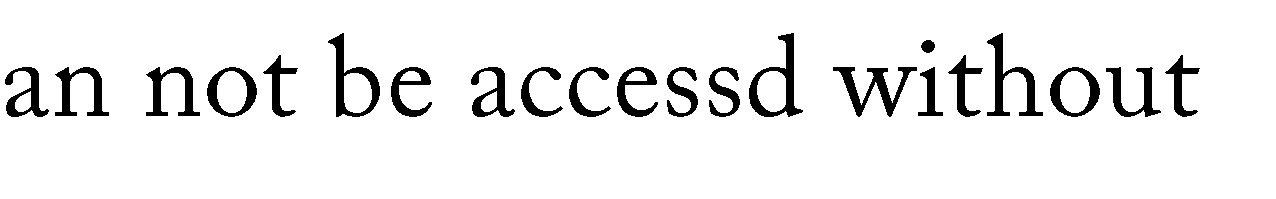
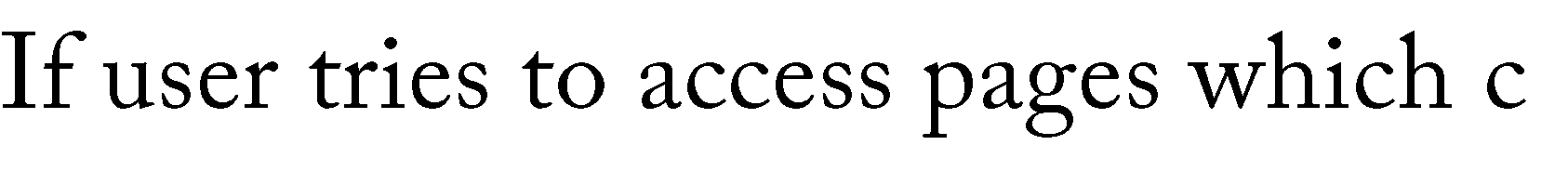


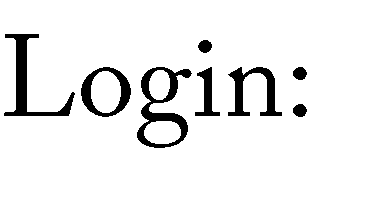
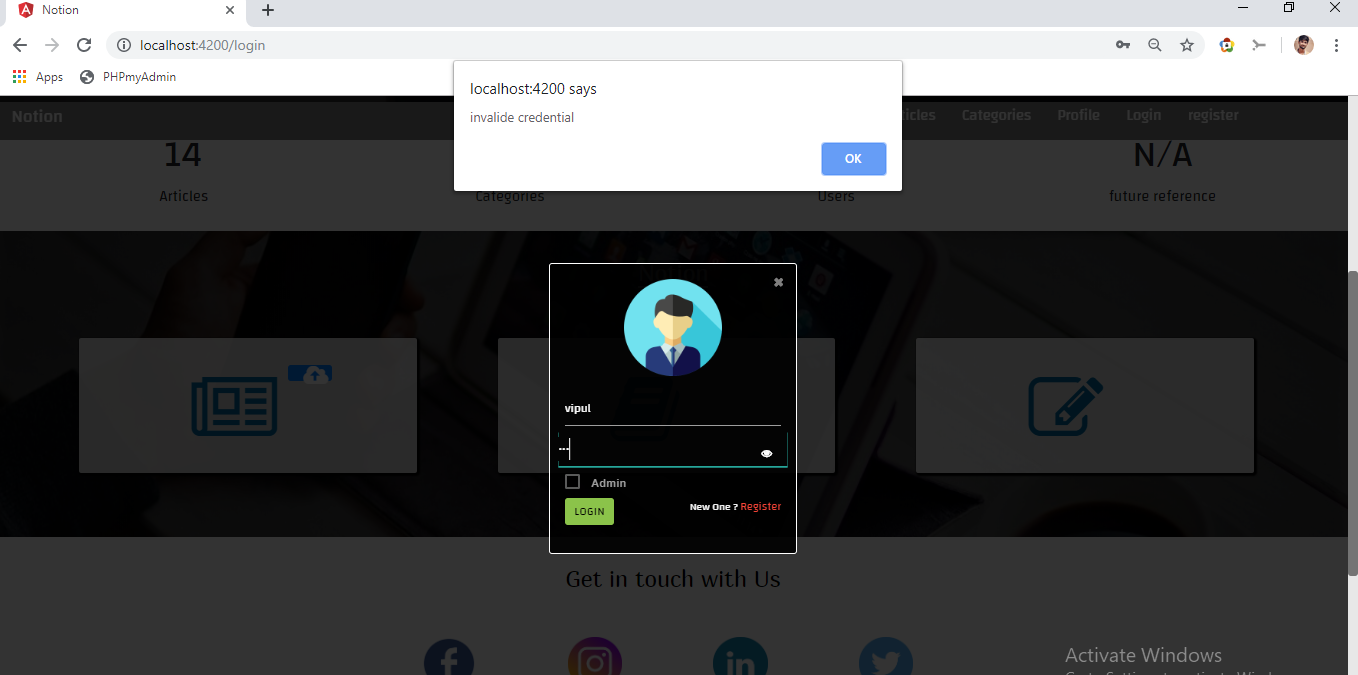


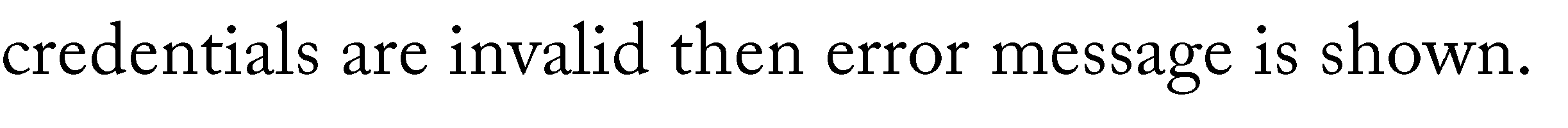
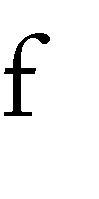
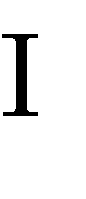


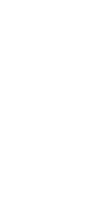
Both Password Must Be Same

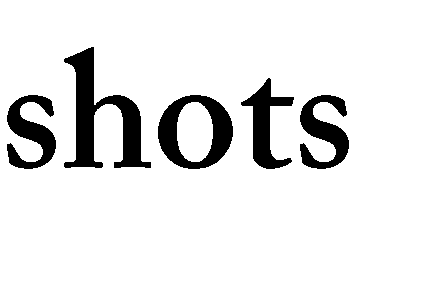
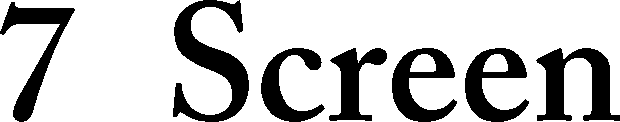


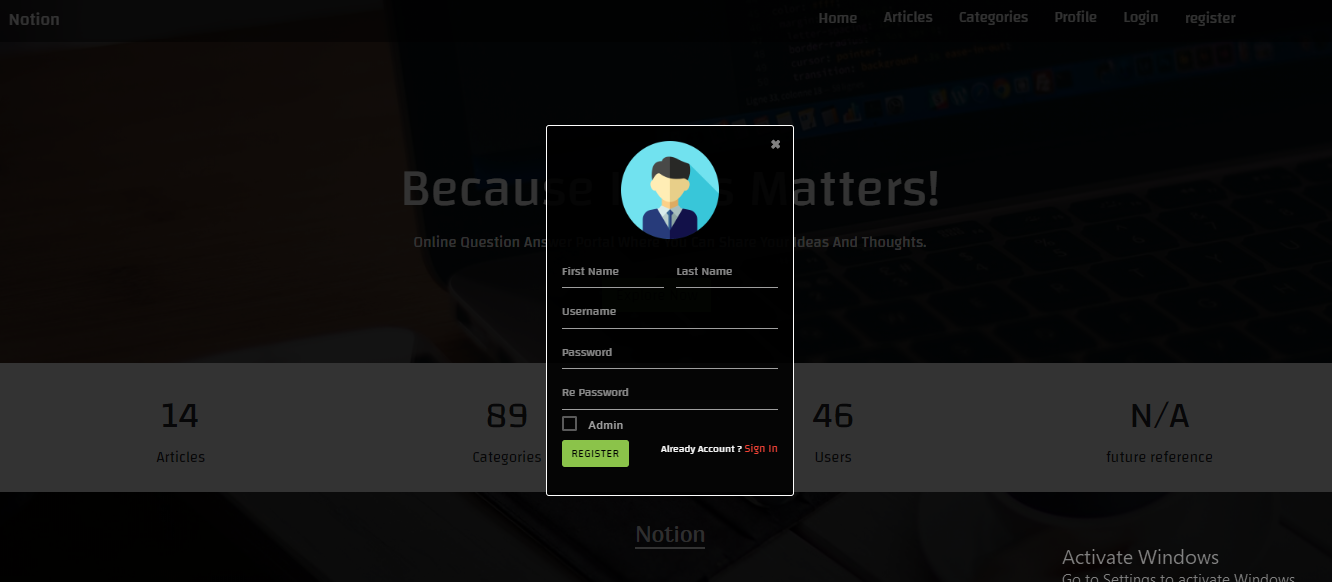


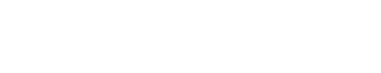










 Figure7.1: Registration

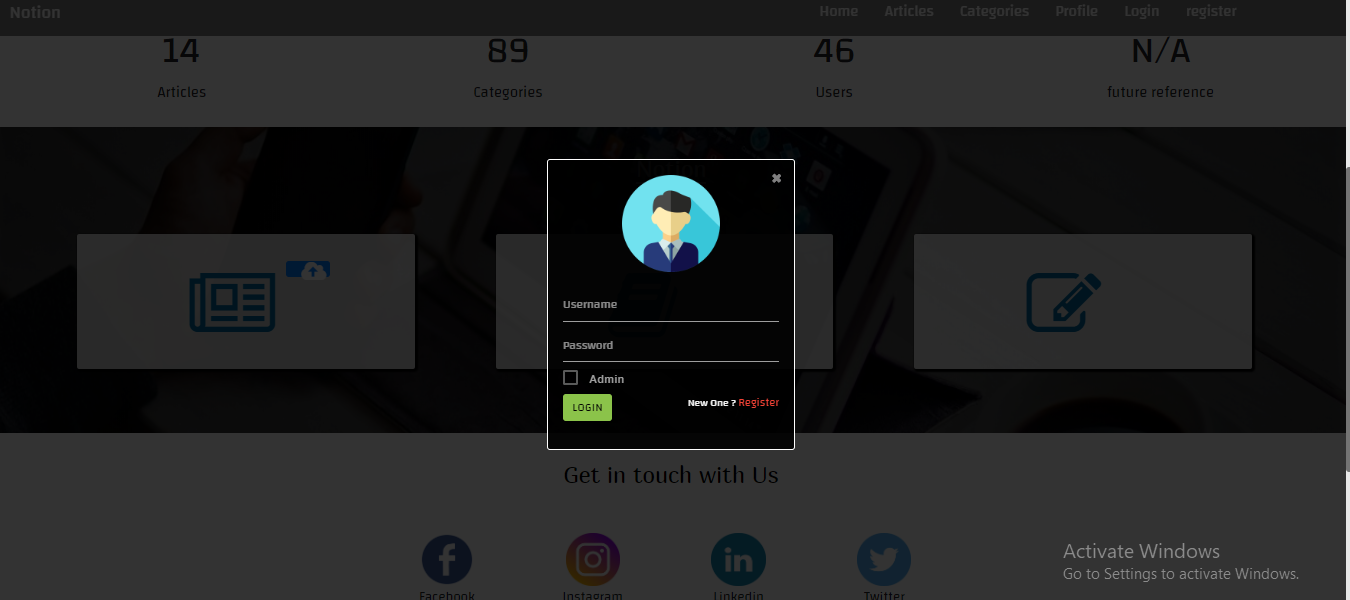
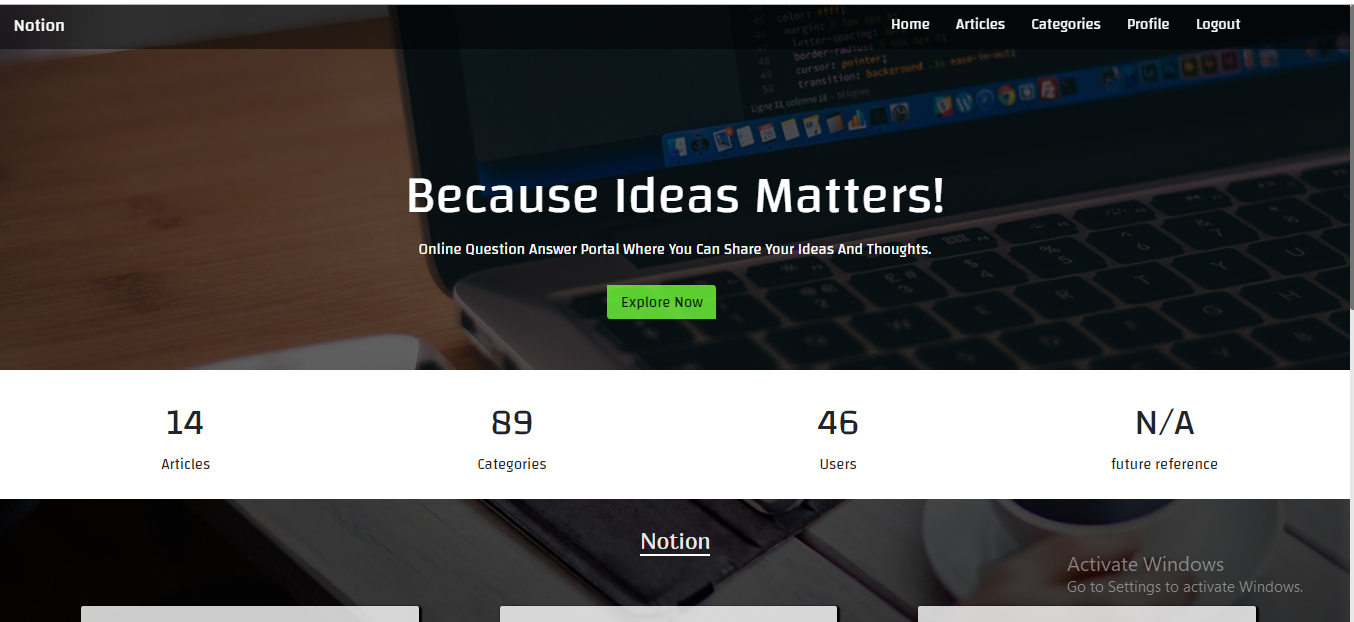


Figure7.2: Signup/Login



7.3 Home page after login

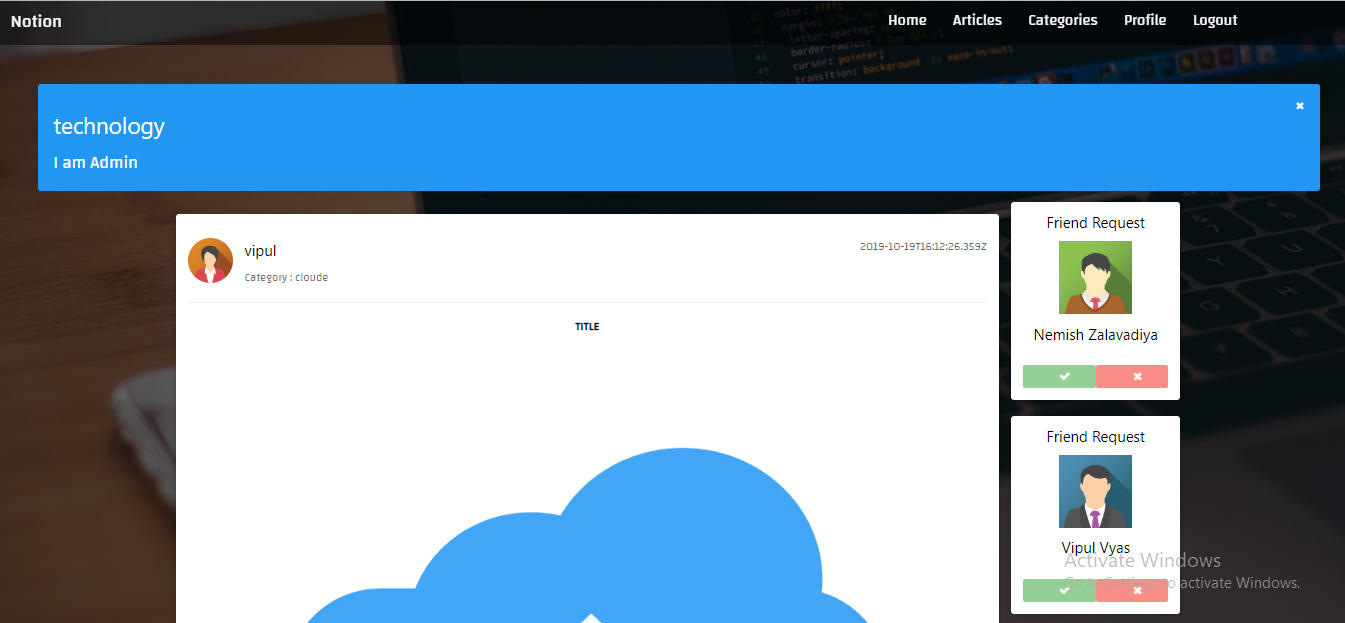


Figure7.4: Read Article

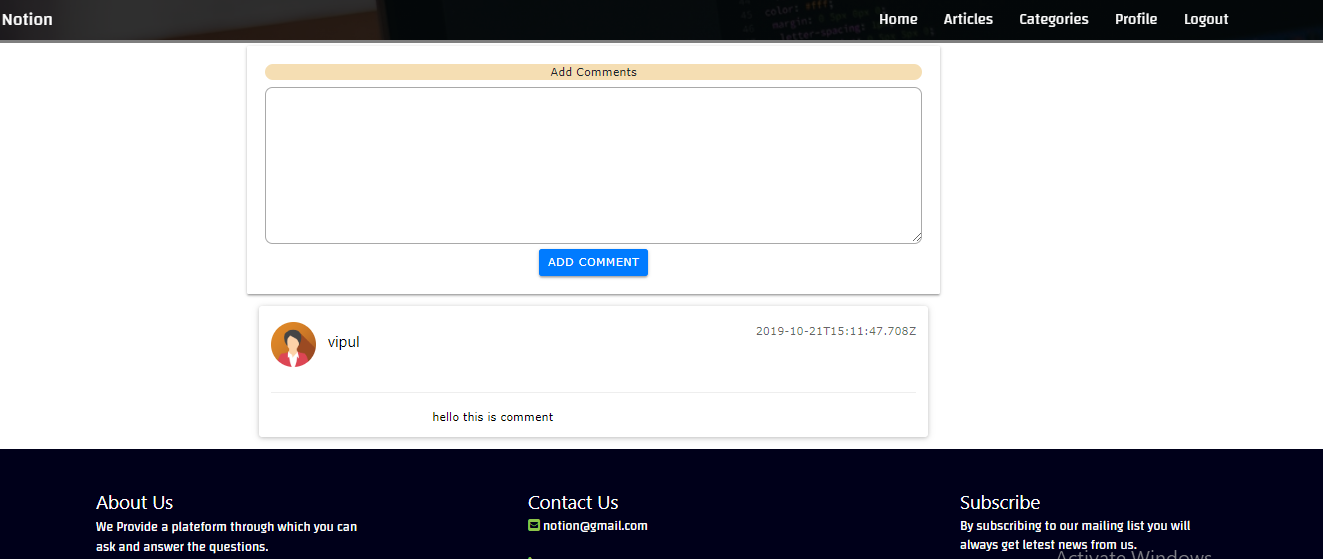


Figure7.5: Article comment

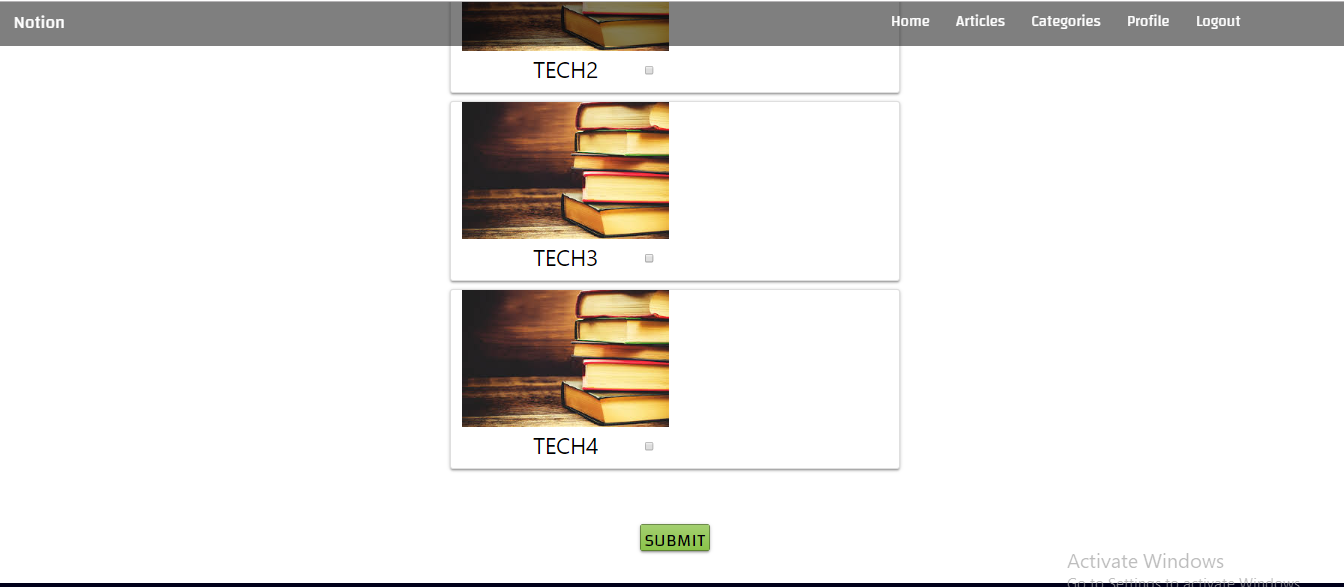


Figure7.6: Choose Interest Category

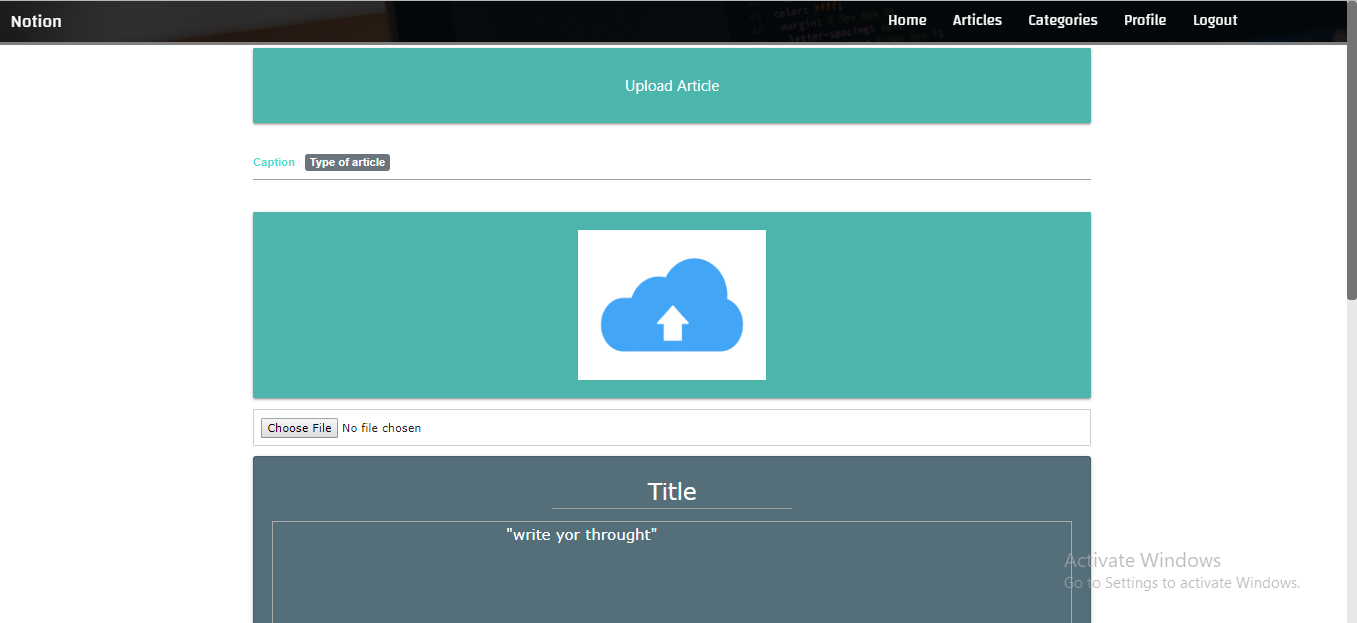
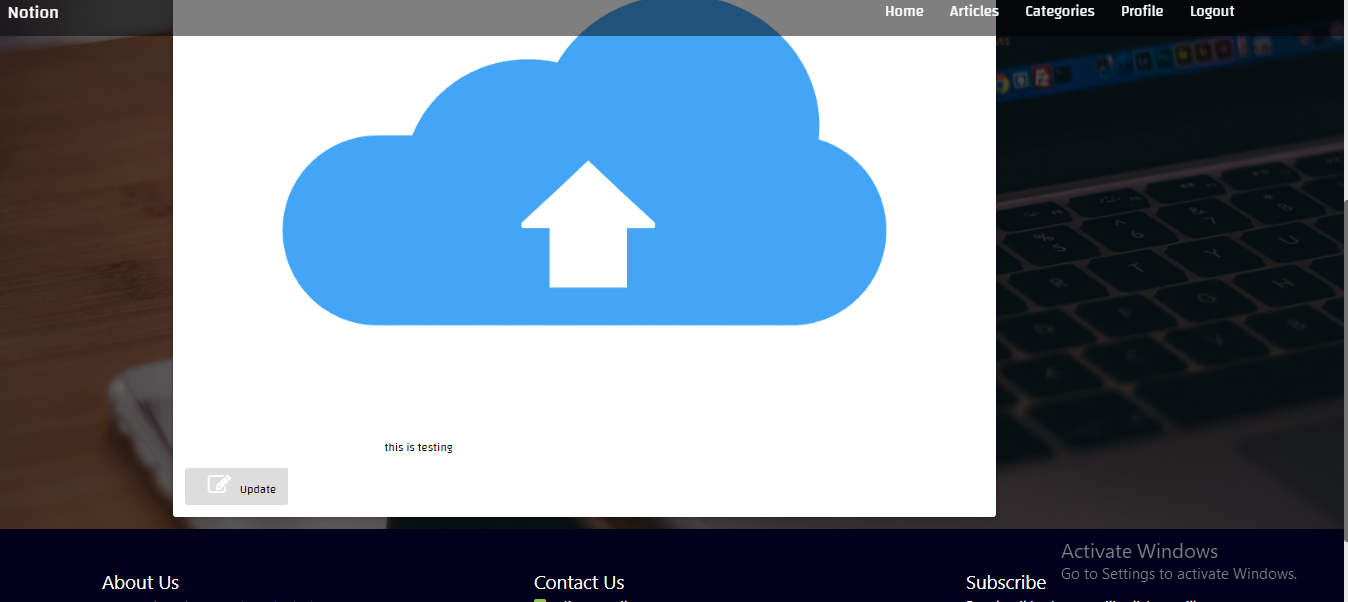
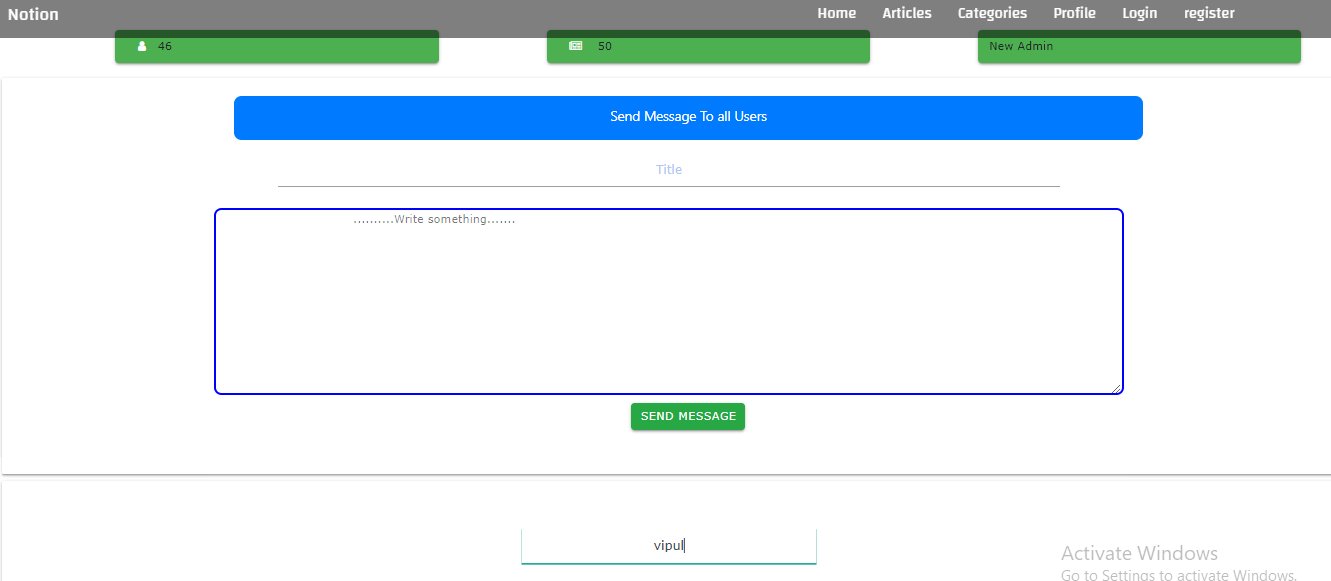


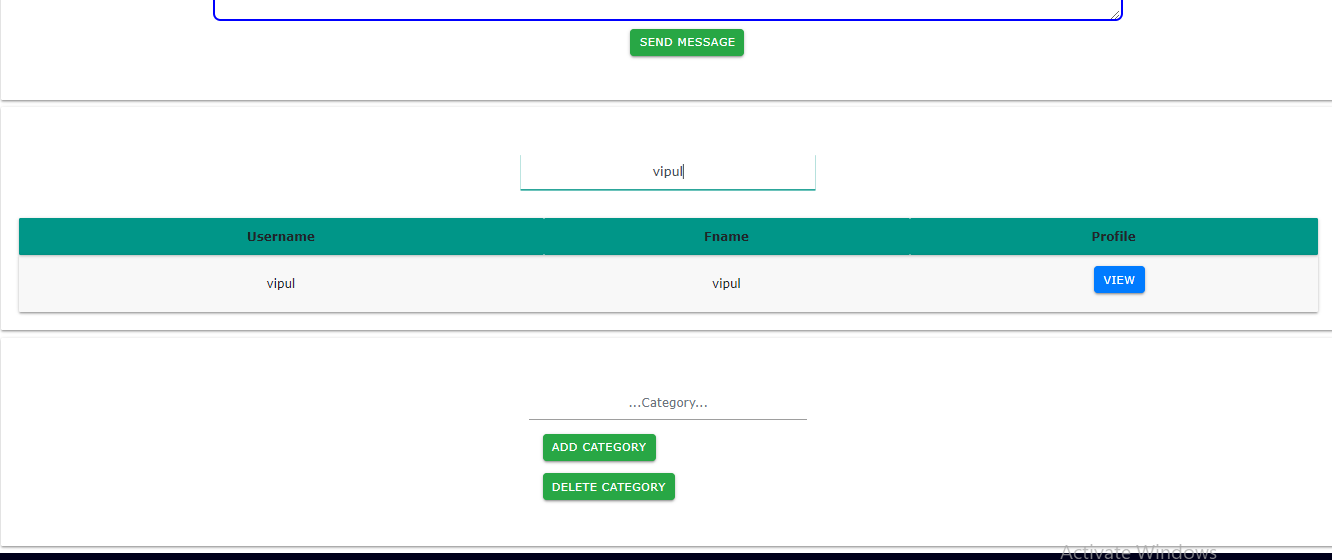
Figure7.7:Upload Article



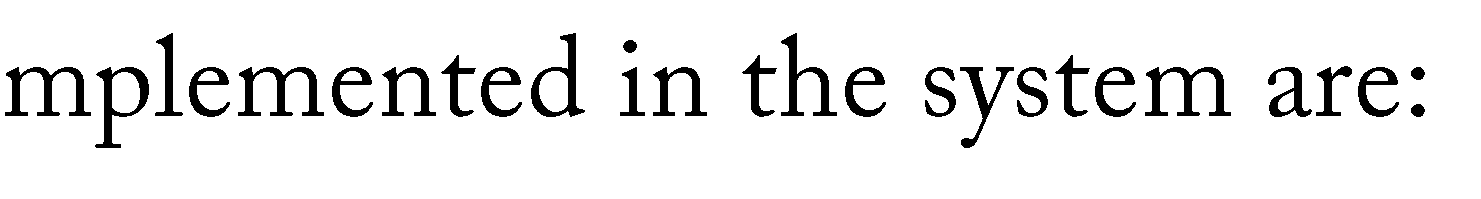
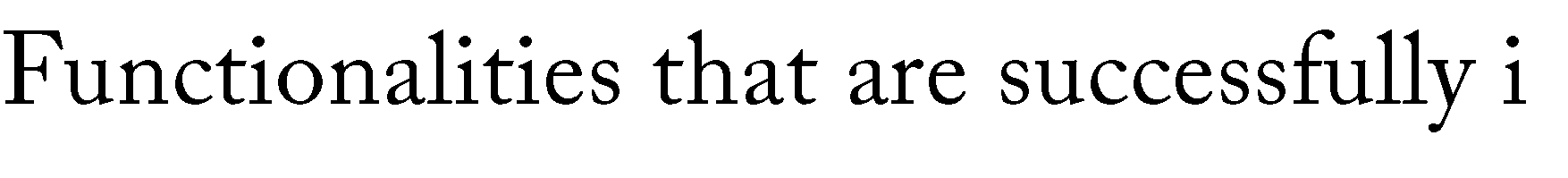
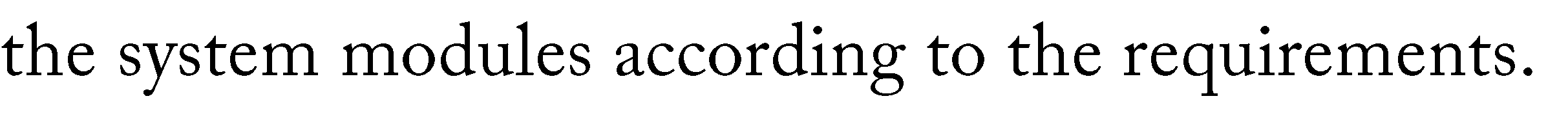
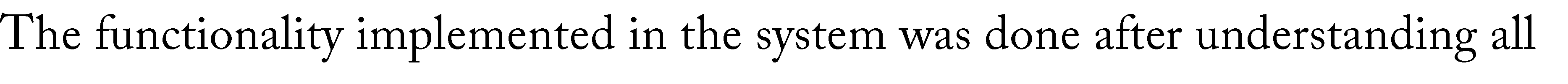
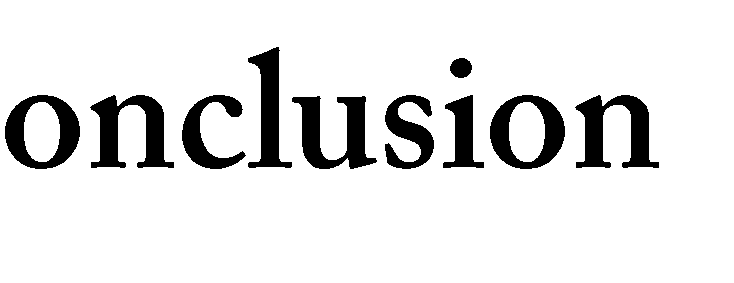
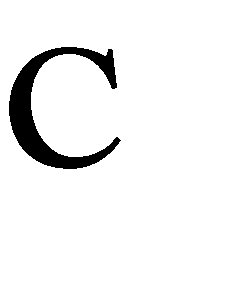
7.8 Update Article



7.9 Admin Panel



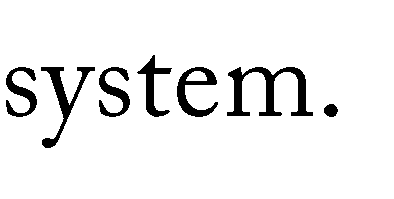
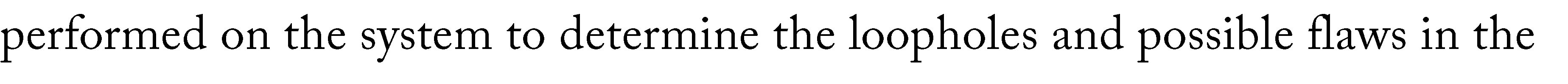
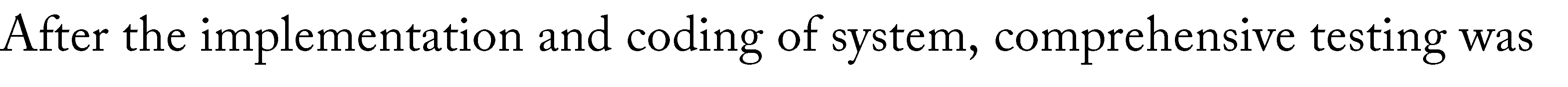
7.10 Search User and Edit Category

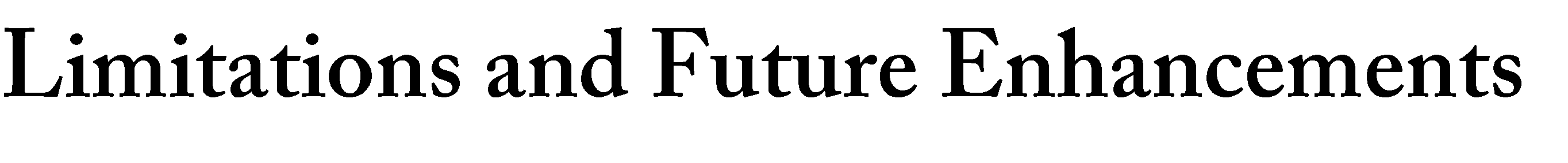


User Registration & Login

User page Authentication

* Article Upload
* Article Editing
* Article Comments
* Upload Comments
* Change Categories
* Send Message to all Users
* Read Articles
* Upload articles with the images
* Search by username

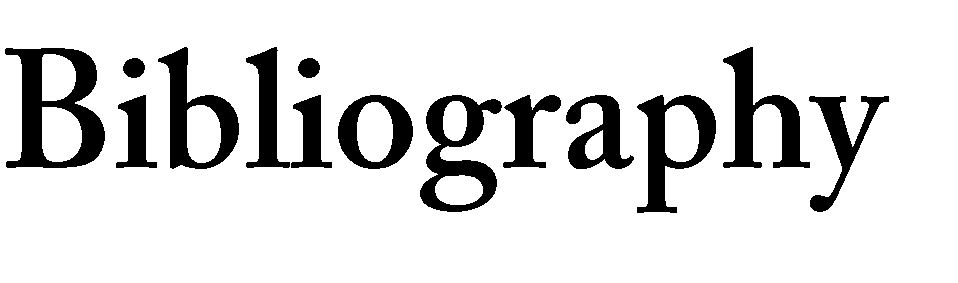




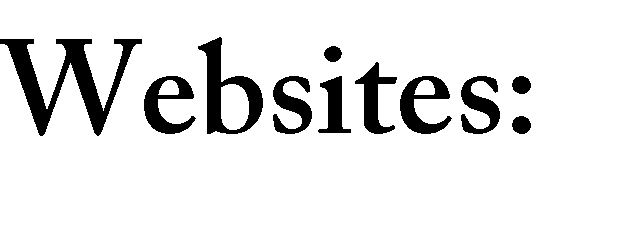
* + - **Limitations**
* nly one user per system
* Photo size and quality must be match with cors policy
* Comment can’t be deleted
  + - **Functionalities not Implemented**
      * User Profile change
        + - password , personnel data
      * Delete Articles, Comments
      * Mailing System
        + Node-mailer
      * Follower & Following
    - **Future Extension**

Following & Follower

* Video upload
* Mailing system
* Article searching
* Article & Comment deleting
* More quality image uploading

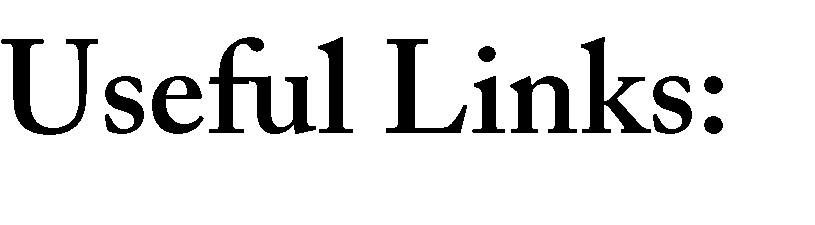






1. [https://www.w3schools.com](https://www.w3schools.com/) –For Html,CSS,Bootstrap,XML,Javascript.
2. https://[www.tutorialspoint.com](http://www.tutorialspoint.com/)‐For NodeJS,TypeScript,AngularJs.
3. [https://angular.io](https://angular.io/) –For Understanding Angular.
4. [https://stackoverflow.com](https://stackoverflow.com/) –For Solving problems.





1. https://nodejs.org/en/ ‐For download and setup of NodeJS.
2. https://fonts.google.com –For getting fonts families.
3. [https://www.youtube.com/watch?v=1tRLveSyNz8&t=4413s](https://www.youtube.com/watch?v=1tRLveSyNz8&amp;t=4413s) – For understanding.
4. [https://www.youtube.com/watch?v=MPuXl1DS1vU&t=235s](https://www.youtube.com/watch?v=MPuXl1DS1vU&amp;t=235s)
5. <https://materializecss.com/> - to get css effects