Department of Computer Science

Final Term Project Proposal, Spring 19-20

|  |  |  |  |
| --- | --- | --- | --- |
| Course Title | ADVANCED PROGRAMMING IN WEB TECHNOLOGY | Section | B |

Group Members:

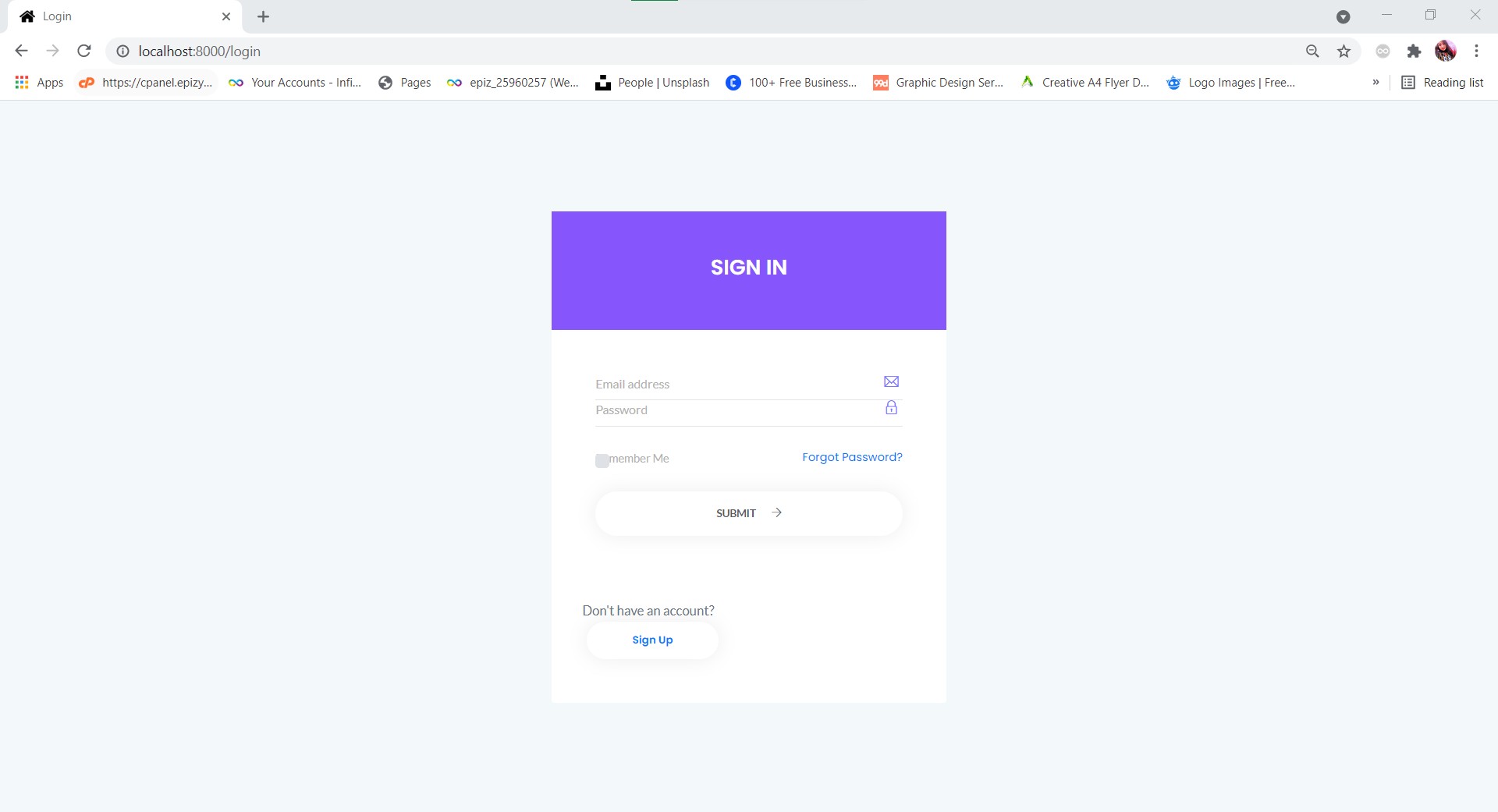
|  |  |  |
| --- | --- | --- |
| **Student ID** | **Name** | **Signature** |
| 17-35507-3 | PRITHA, JUI SAHA (Admin) |  |
| 18-36939-1 | S. M. HASIB ( Manager) |  |
| 18-36304-1 | RAJAN DAS GUPTA (Creator) |  |
| 18-36933-1 | SHOURAV SHOME (User) |  |

**Project Title: Content Management System**

Short Description:

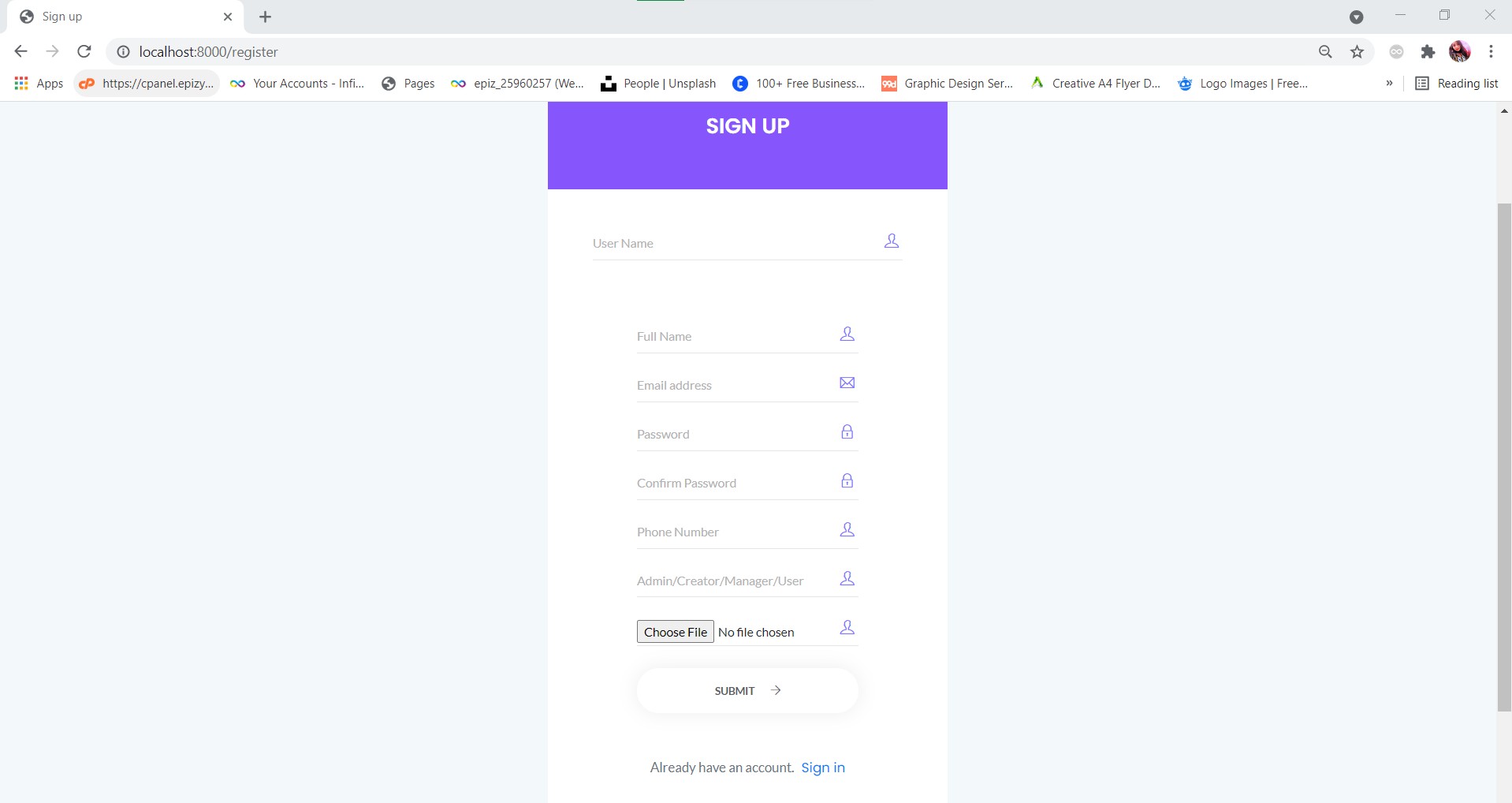
|  |  |
| --- | --- |
| Role Name | Description |
| **Admin** | Admin can login/logout to the system and registration as a admin by giving proper information. Admin Can view/add/delete/modify all Manager and Creators profile. Admin also modify his own profile. Admin view all contents from the creator and view all oder status from manager. In addition, admin can add notice and delete notice . |
| **Manager** | Content manager can login/logout to the system and registration for users by giving proper information. As manager he can approve content post request from from content creator. he can view/delete/modify the content price, order approval order list, Content informations. Can update own profile. |
| **Creator** | Creators can login/logout to the system and registration as a creator by giving proper information. As a Creator he can view/add/delete/edit all contents like image, video, PDF & also contents information. Can also view all content list. Can update own profile. He can also view order page’s information. Creator can see list of order ratings and review about contents. |
| **User** | Users can login to the system and view all contents from the creator. User can also see list of all creator, manager and admin with their details by scanning a QR code. A user can order for a custom content by uploading his own content related template or image. He can check status if his order is at pending or accepted by creator. While buying any content from a creator he has his own add to cart option. And also a wishlist option to keep track of his favourite content to buy later. And user can download free contents from creator. User can also edit his own profile and can delete or edit orders created by him. User can also give ratings to a content and write review about it. |

Modules:



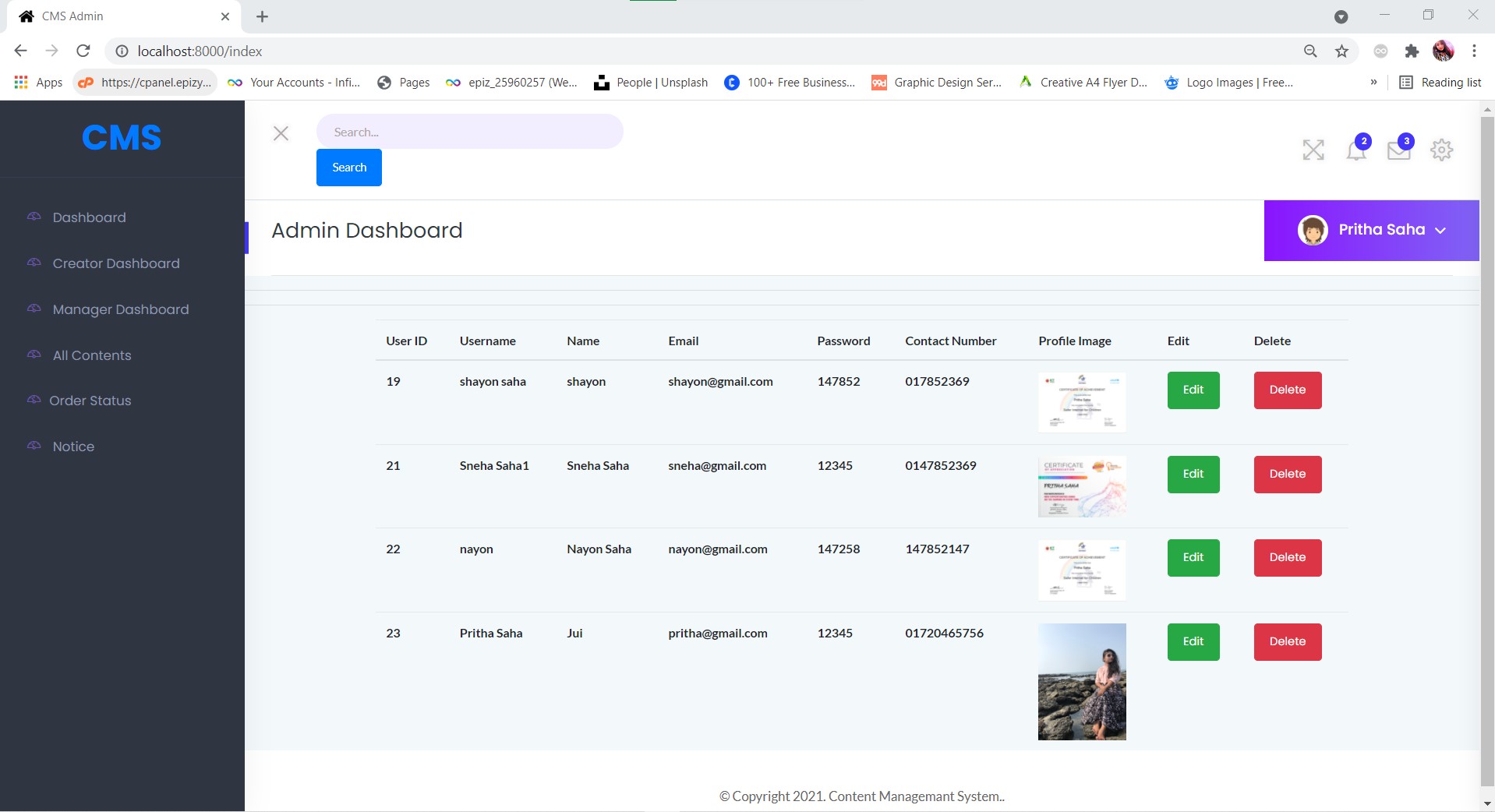
|  |  |  |
| --- | --- | --- |
| **Login Module (M1)** | |  |
| **Function** | Description | Database |
| **M1F1** | In the view of of **Login page** in the route of **“/login”** and there’ll be a html form having fields **email, password,**and a login button to fire form submission to the same route with post request. | N/A |
| **M1F2** | After receiving post request following form validation will be happen,  Validates using Laravel Illuminate\Supports\Validator   * Email validation. * Empty field validation. * Email existence validation (From database users table). | Match data from db table **“users”**  and columns **email .** |
| **M1F3** | PHP Manual validation for password cross checking (From database users table). | Match data from db table **“users”**  and columns **email** and **password.** |
| **M1F4** | If everything is fine **email, password** will be saved in the database. After then user will redirect to the system. | N/A |

|  |  |  |
| --- | --- | --- |
| **Logout Module (M2)** | |  |
| **Function** | Description | Database |
| **M2F1** | There have logout link in system header after clicking logout button the authentication session will remove and user redirect login page**.** | N/A |

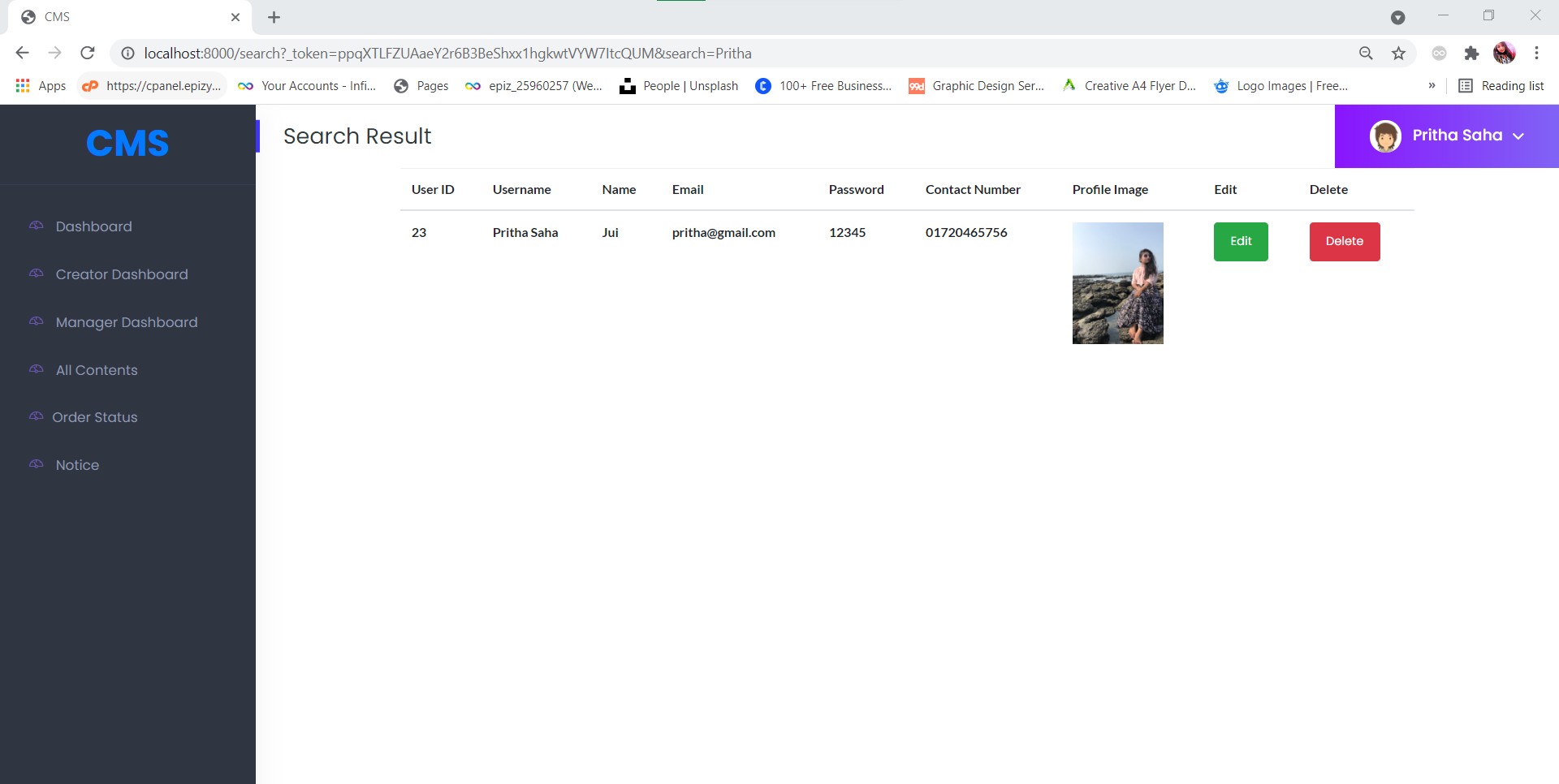


|  |  |  |
| --- | --- | --- |
| **Registration Module (M3)** | |  |
| **Function** | Description | Database |
| **M3F1** | In the view of login page a Signup Button will be added referring the route **“/register”** | N/A |
| **M3F2** | In the view there’ll be a html form having fields **name, email, password, password\_confirmation , company** and a submit button to fire form submission to the same route with post request. | N/A |

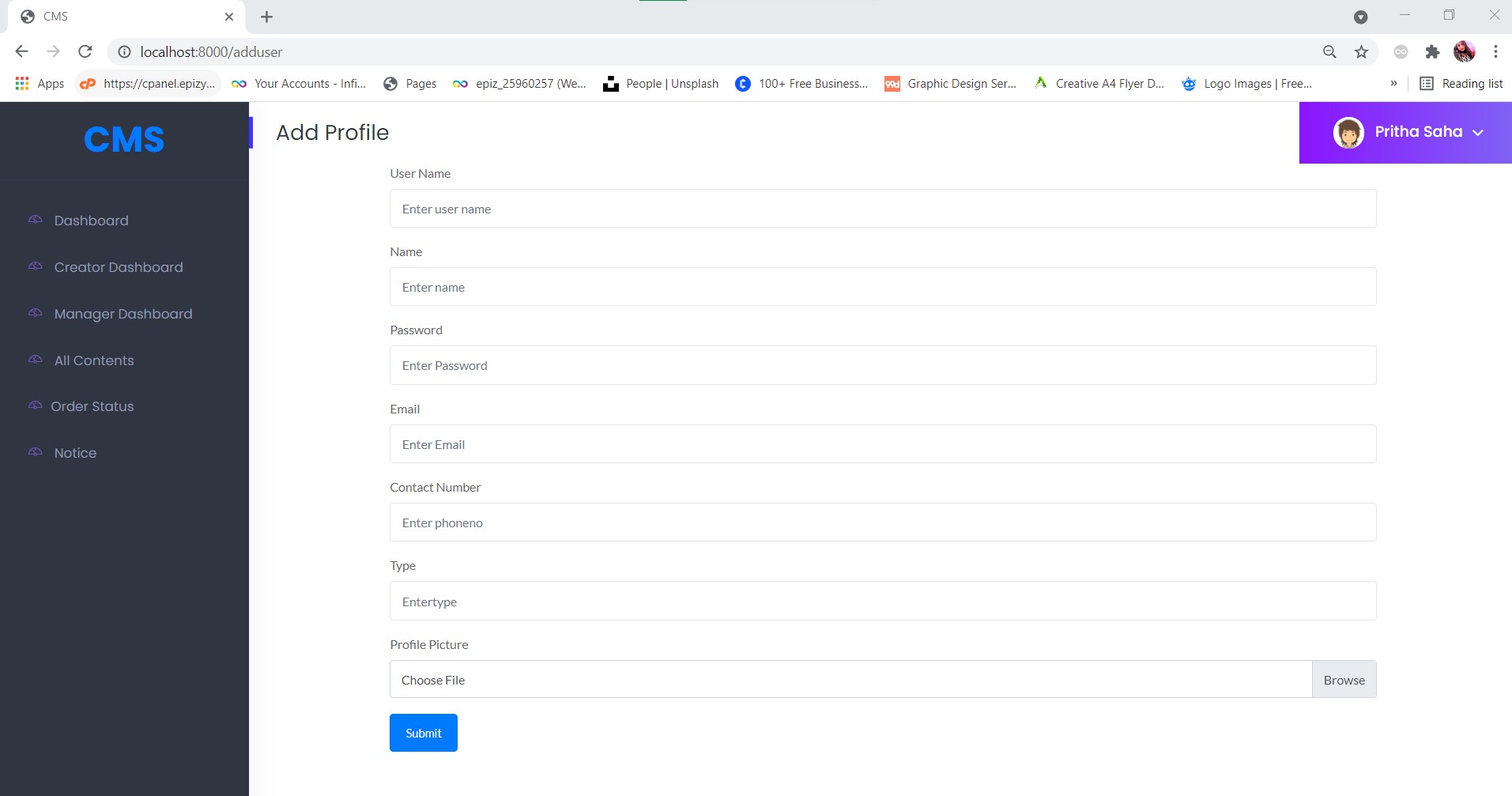
|  |  |  |
| --- | --- | --- |
| **M3F3** | Validates using Laravel Illuminate\Supports\Validator   * Empty field validation * Email validation * Email and Company existence validation (From database users table)   Password and Confirmed password validation  If everything is fine user**name**,**name, email, password, type, profilepic** will be stored in the database. After then user will redirect to **login page.** | Match data from db table **“users”** and columns user**name**,**name, email** and **type.**  Stores **name=name, email=email, password=password**,, validated = 1, type =admin,  operator=null to the db table  **“users”** |
| **M3F4** | After successful registration, the email and name store in session and redirect to loginpage with successful message by name with fill up the email field. | N/A |



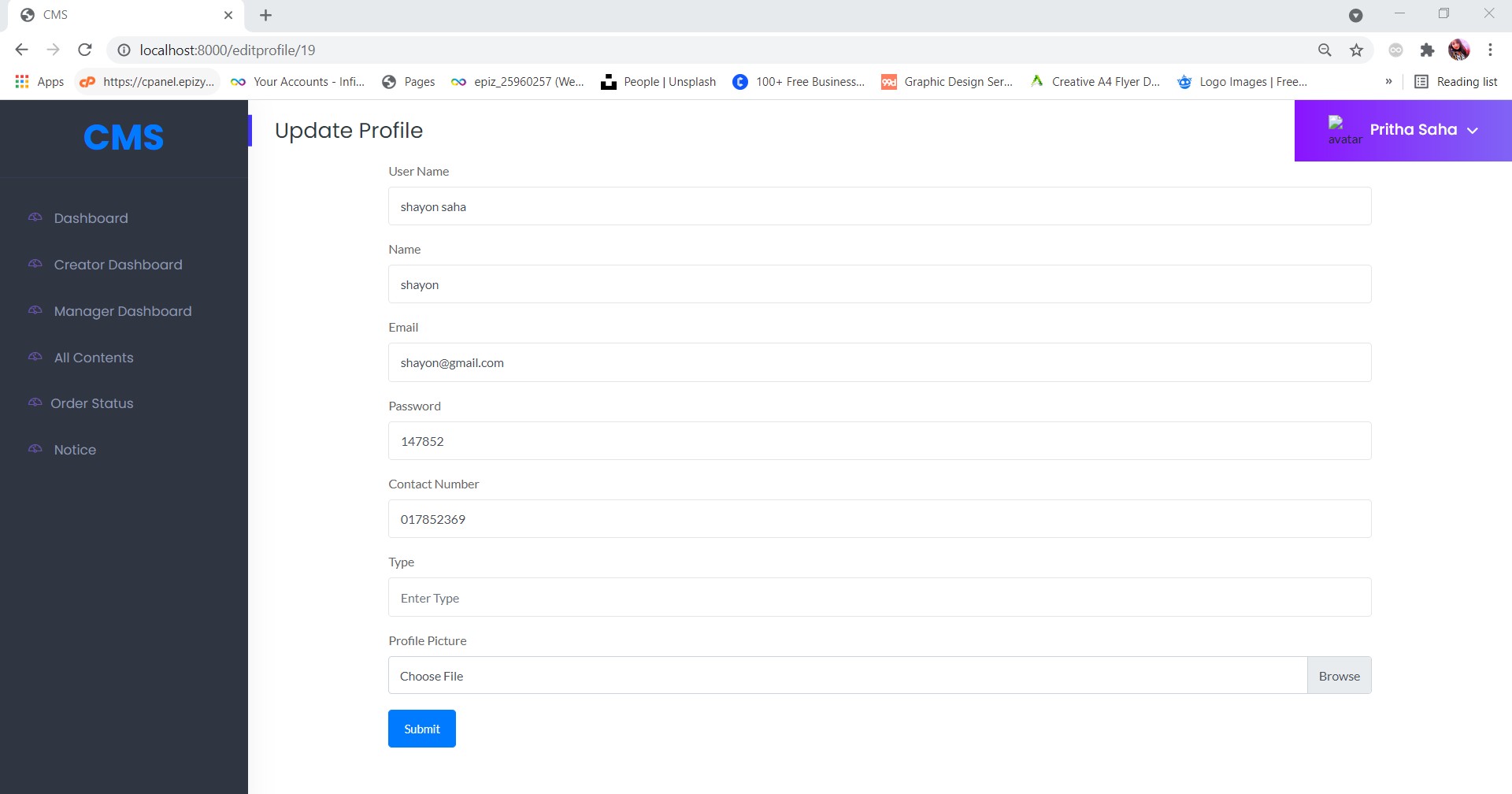
|  |  |  |
| --- | --- | --- |
| **Admin Listing Module (M4)** | |  |
| **Function** | Description | Database |
| **M4F1** | **Admin’s** listing view can be access by route **“/index”** . In the view there will be a table having column **userid, username, name, email, password, contact no, profile image, action**. All the respective data will be shown from database. Action column will be use to show action buttons such as **Edit, Delete** etc. | List will use db table **“users”** and columns **userid, username, name, email, password, phone no, type, profile image** |



|  |  |  |
| --- | --- | --- |
| **Admin Searching Module (M5)** | |  |
| **Function** | Description | Database |
| **M5F1** | It will use the view of **Dashboard** to list the search result and can be access by same route. There’ll be a search field named **“search”** and a search submit button to fire the searching event**.** | List will use db table **“users”** and columns **userid, username, name, email, password, phone no, type, profile image** |
| **M5F2** | Search event will call for an **Ajax (post) request** to **‘ajax/search’** with the parameter **search** having value of search field. Once the search result is received then it will list on **Dashboard Listing.** | Matches the search parameter with the db table **“users”** and columns **name, email, type.** Matching relation will be **OR** and pattern will be **LIKE %%** |

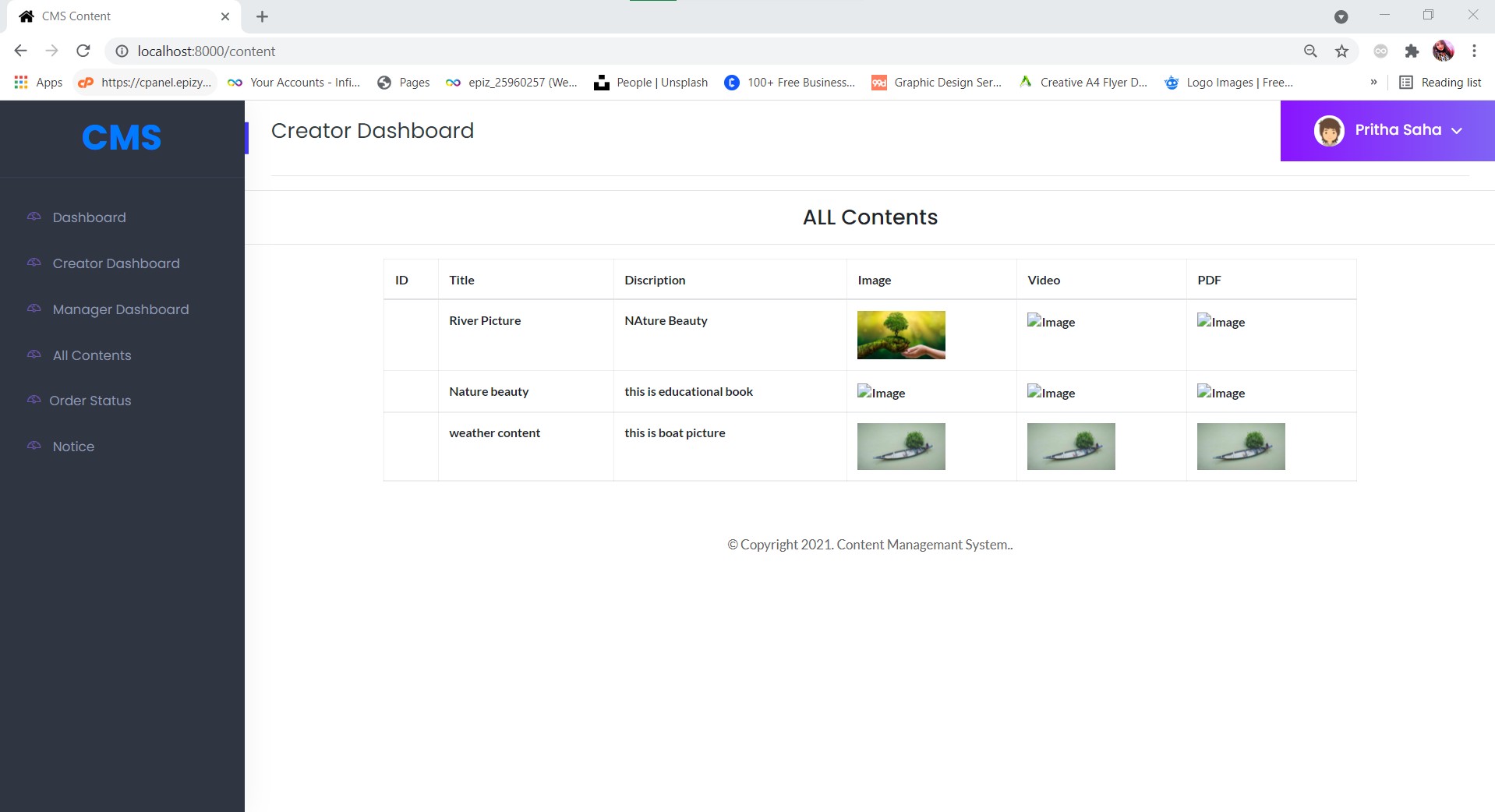


|  |  |  |
| --- | --- | --- |
| **Admin Add (M8)** | |  |
| **Functio n** | Description | Database |
| **M8F1** | In the view of Module **Dashboard** a Add New Button will be added referring the route **“/adduser”** | N/A |
| **M8F2** | In the view there’ll be a html form having fields **userid, username, name, email, password, contact no, profile image** and a submit button to fire form submission to the same route with post request. | N/A |
| **M8F3** | After receiving post request following form validation will be happen,   * Name: Required * Email: Required | Valid Email | Unique * Password: Required | Length:4| Confirm   If everything is fine **name, email, password** will be stored in the database. After then user will redirect to **Dashboard.** | Stores **name=name, email=email, password=password**, validated = 1, role =creator, type=null, operator=null to the db table **“users”** |

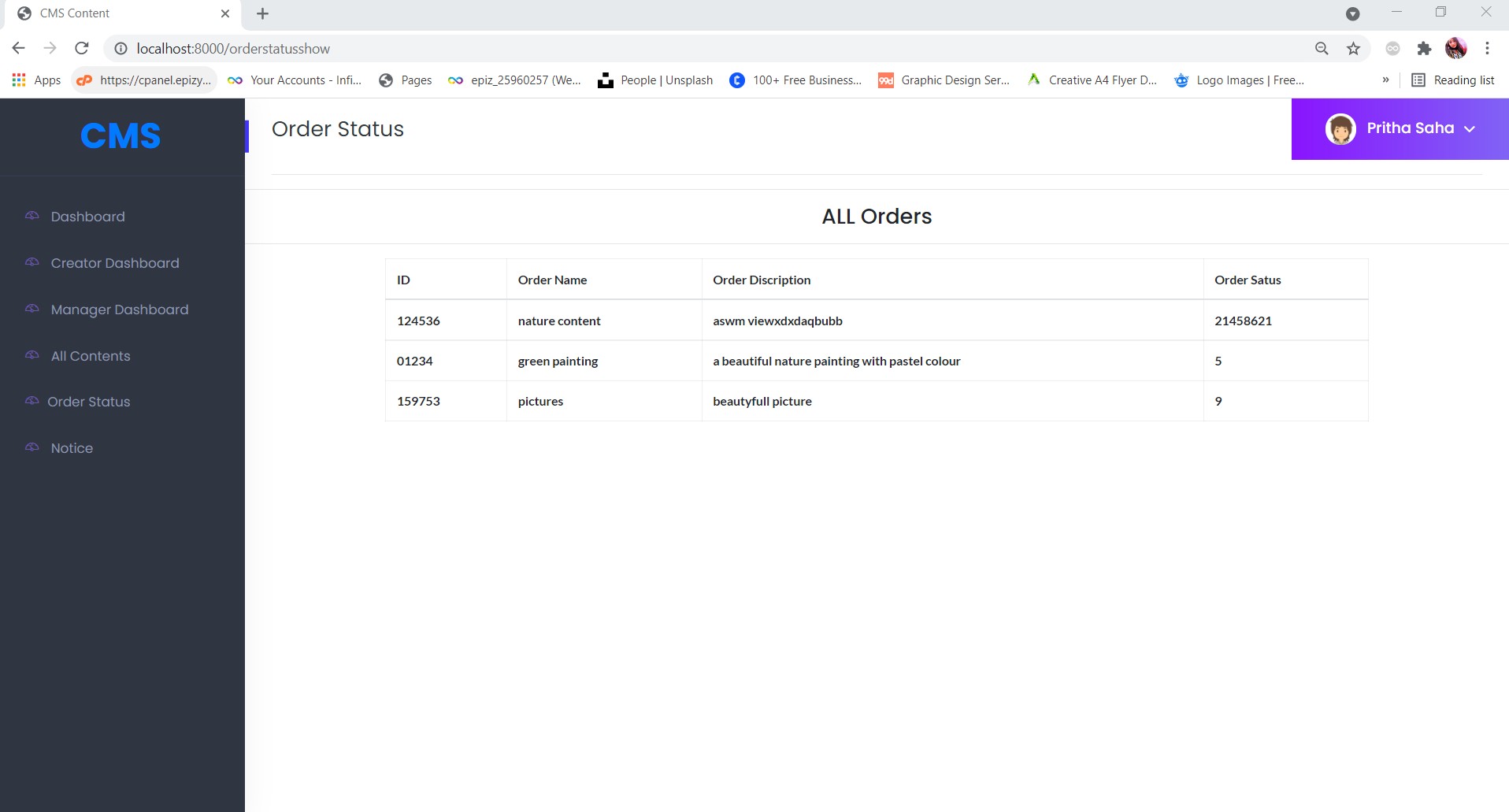


|  |  |  |
| --- | --- | --- |
| **Admin Edit (M6)** | |  |
| **Function** | Description | Database |
| **M6F1** | In the List of Module **Dashboard** a **Edit** Button will be added to every row referring the route **“/editprofile/{id}”.** Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“users”** |
| **M6F2** | In the view there’ll be a html form having fields **username, name, email, password, phone no, type, profile image** and a submit button to fire form submission to the same route with **Post** request. | N/A |
| **M6F3** | After receiving **Post** request support staff will be retrieved from the database and following form validation will be happen, | Find User from db table **“users”**  by **“id”** |
|  | * Name : Required * Email: Required | Valid Email | Unique except Current * Password: Nullable | Length:4   If everything is fine **name, email, password,company** will be saved in the database. After then user will redirect to dashboard. | Updates **name=name, email=email, password=password,**  **type=type** to the db table  **“users”** |

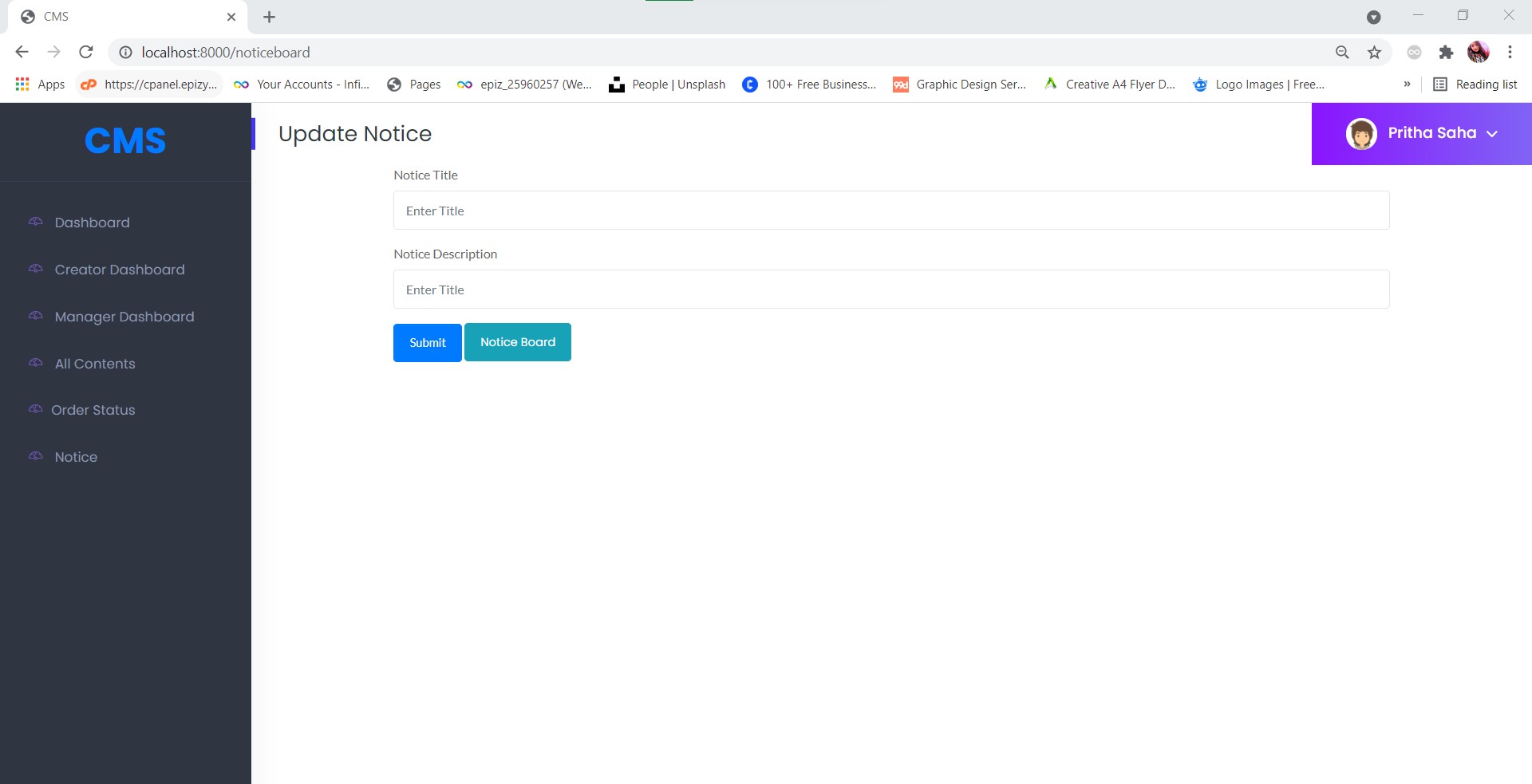
|  |  |  |
| --- | --- | --- |
| **Admin Delete (M7)** | |  |
| **Function** | Description | Database |
| **M7F1** | In the List of Module **dashboard** a **Delete** Button will be added to every row referring the route **“/delete/{id}”.** Here id parameter holds the admin id of a row. | Uses the value of **“id”** column of the db table **“users”** |
| **M7F2** | In the view there’ll be a javascript confirmation**.** After confirm a virtual form will be submitted to the route of **dashboard** on **Delete** Request. | N/A |
| **M7F3** | After receiving delete request admin will be retrieved from the database and will be removed. After then user will redirect to **dashboard.** | Find User from db table **“users”**  by **“id”**  Delete from db table **“users”** by  **“id”** |



|  |  |  |
| --- | --- | --- |
| **All Content Listing Module - Admin (M9)** | |  |
| **Function** | Description | Database |
| **M9F1** | **Content’s** listing view can be access by route **“/system/contents”** . In the view there will be a table having column **title, image**, **pdf**, **video**, **description, crating, ccomplain**.. All the respective data will be shown from database. Action column will be use to show action buttons such as **Upload**,**Edit, Delete, View, Download** etc. | List will use db table **“contentcreator”** and columns **id, title, image**, **pdf**, **video**, **description, crating, ccomplain**. |

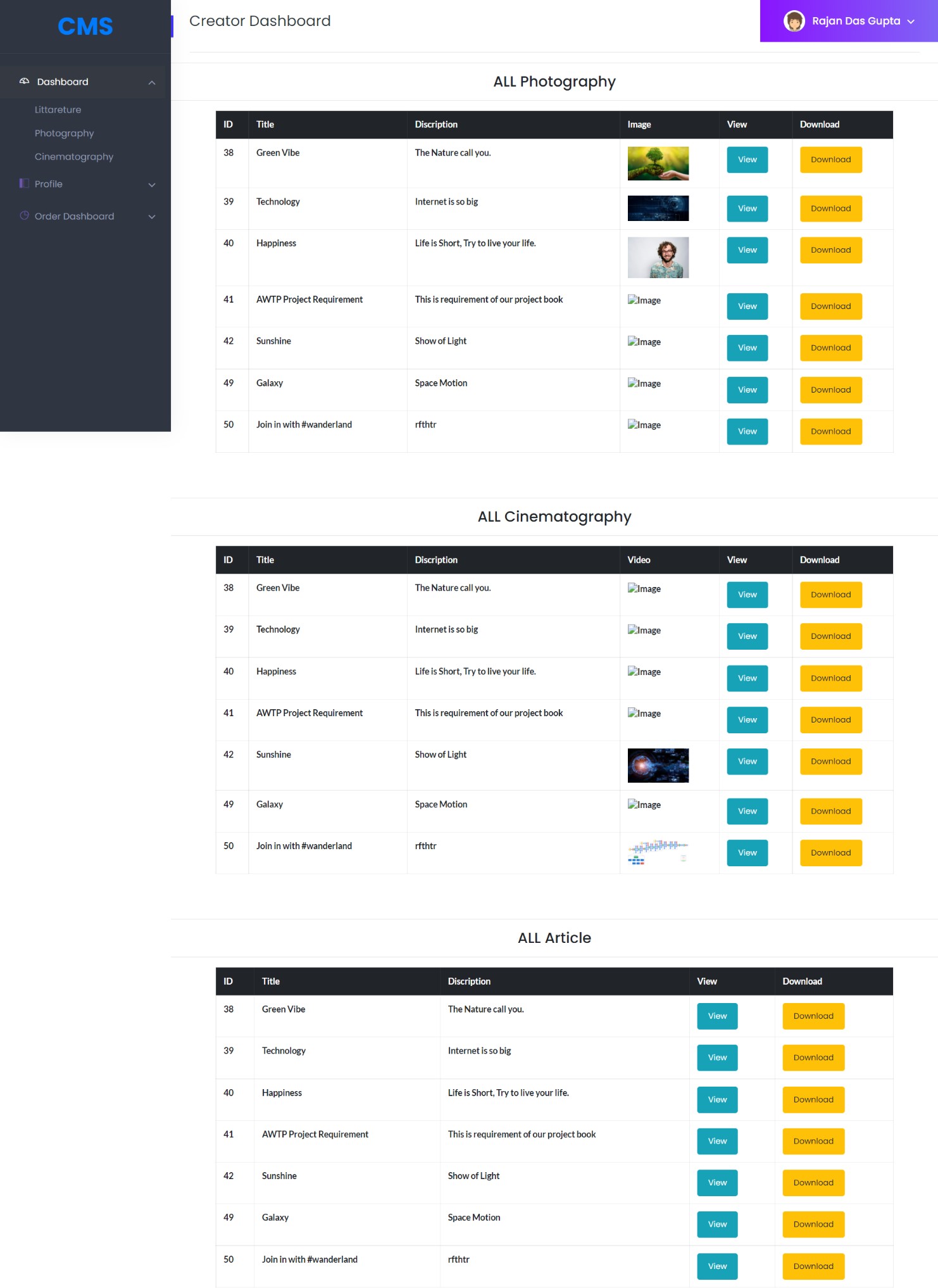


|  |  |  |
| --- | --- | --- |
| **Order Status Module- Admin(M10)** | |  |
| **Function** | Description | Database |
|  | **Order status** view can be accessed by route **“/orderstatusshow”** by clicking **Order Status** in sidemenu of dashboard In the view there will be a table having **Ordername, Order Description, Order Template, Order Image.** List of all available content will be shown. | List will use db table **“orders”.** columns **ordername, orderdescription**, **template**, **picture**. |

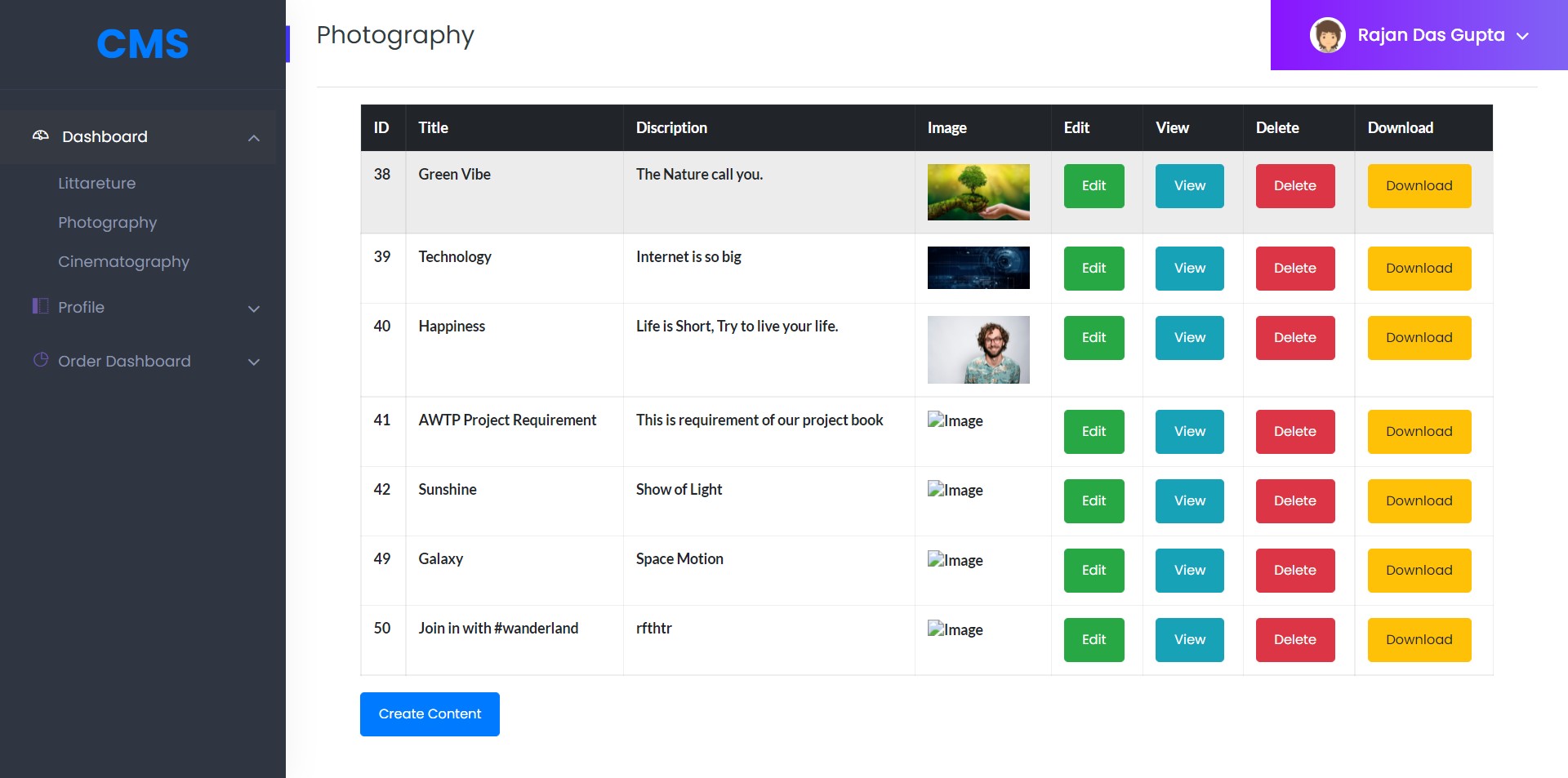


|  |  |  |
| --- | --- | --- |
| **Notice- Admin(M11)** | |  |
| **Function** | Description | Database |
|  | **Notice** view can be accessed by route **“/allnotice”** by | List will use db table |
| clicking Notice in sidemenu of dashboard In the view | “noticestatus”. columns noticeid, |
| there will be a table having**Notice ID, Notice Topic,** | ndescription,topic. |
| **Notice Description.** List of all notice will be shown. |  |

|  |  |  |
| --- | --- | --- |
| **Notice Delete (M12)** | |  |
| **Function** | Description | Database |
| **M12F1** | In the List of **Dashboard** a **Delete** Button will be added to every row referring the route **“/delete/{id}”.** Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“users”** |
| **M12F2** | In the view there’ll be a javascript confirmation**.** After confirm a virtual form will be submitted to the route of **Dashboard** on **Delete** Request | N/A |
| **M12F3** | After receiving delete request support staff will be retrieved from the database and will be removed. After then user will redirect to **Dashboard.** | Find User from db table **“users”**  by **“id”**  Delete from db table **“users”** by  **“id”** |

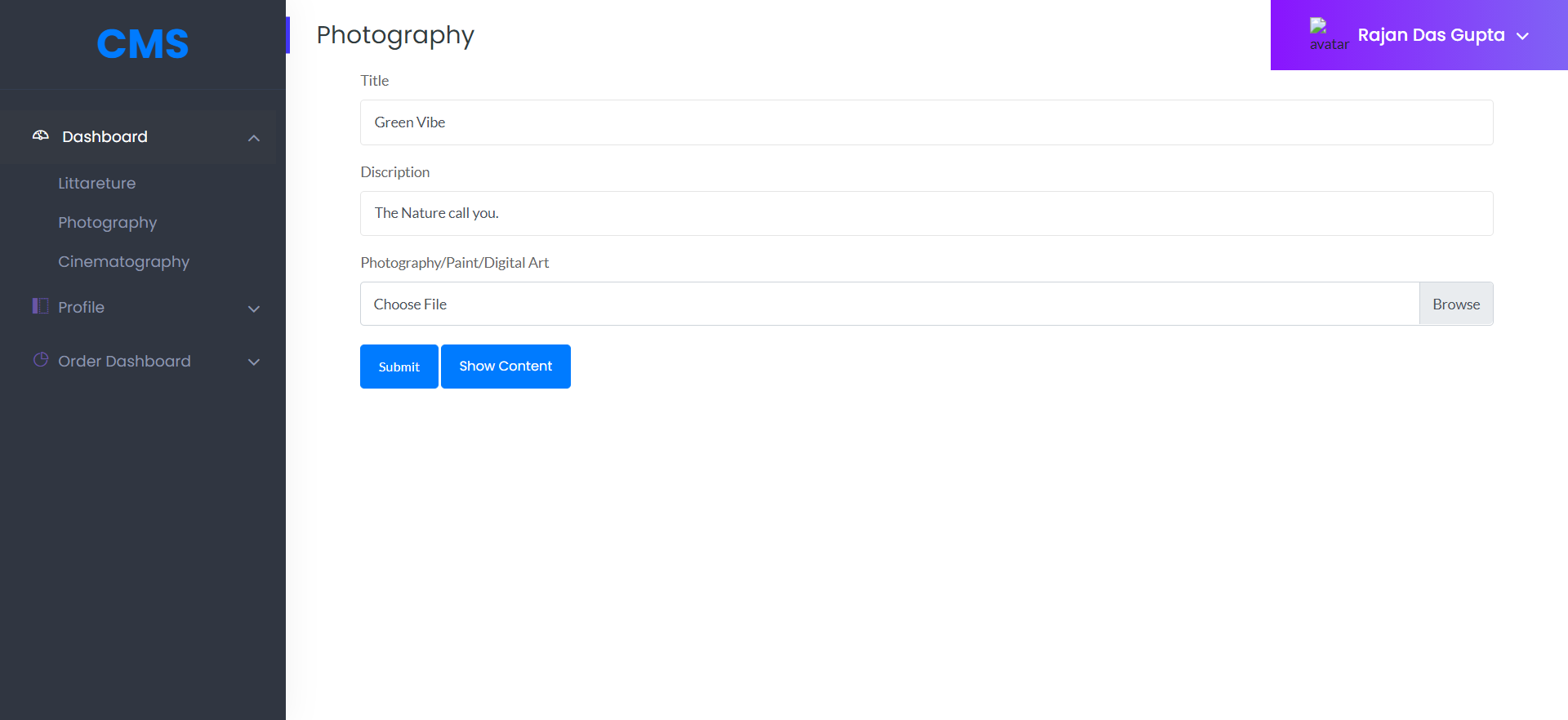


|  |  |  |
| --- | --- | --- |
| **All Content Listing Module - Admin (M13)** | |  |
| **Function** | Description | Database |
| **M13F1** | **Content’s** listing view can be access by route **“/creator/contents”** . In the view there will be a table having column **title, image**, **pdf**, **video**, **description, crating, ccomplain**.. All the respective data will be shown from database. Action column will be use to show action buttons such as **Upload**,**Edit, Delete, View, Download** etc. | List will use db table **“contentcreator”** and columns **id, title, image**, **pdf**, **video**, **description, crating, ccomplain**. |

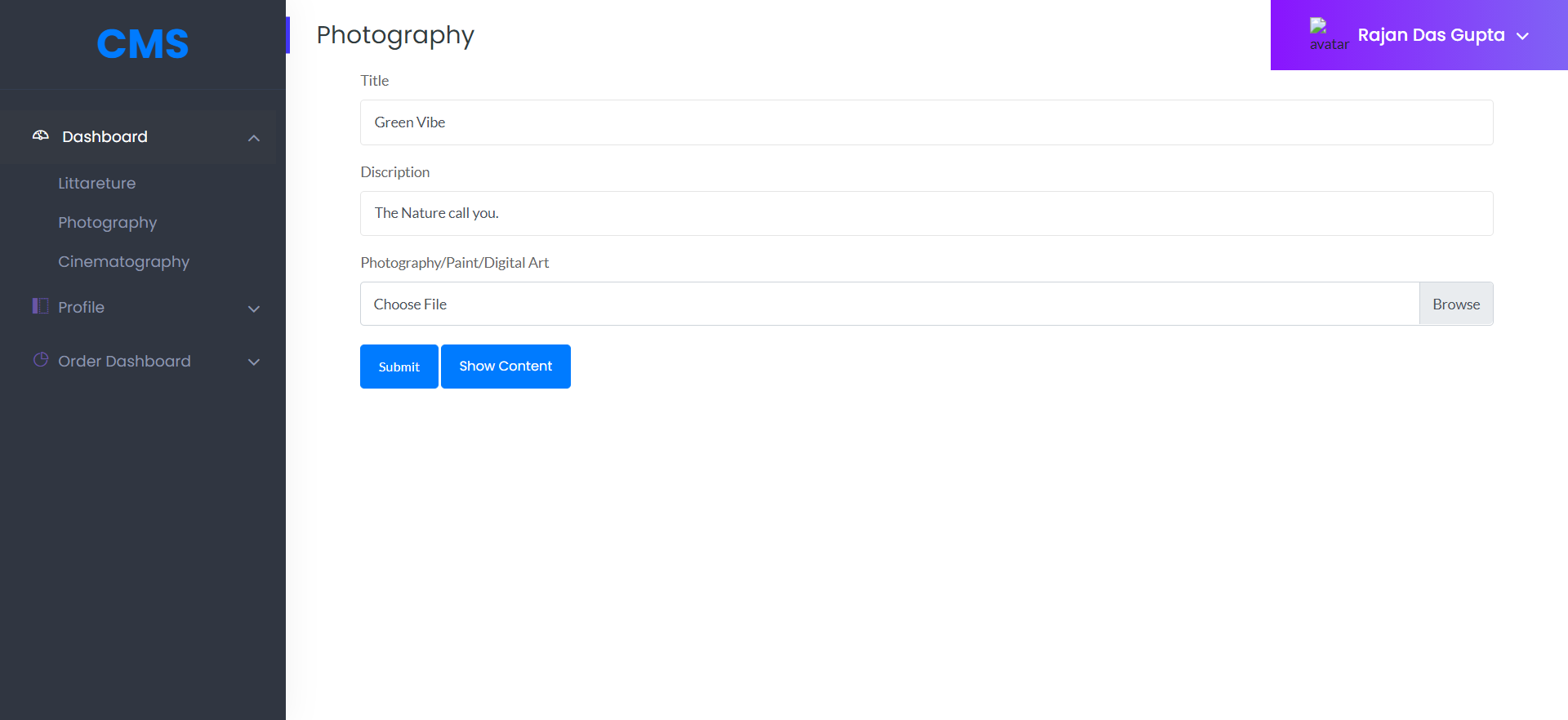


|  |  |  |
| --- | --- | --- |
| **Image/Video/Pdf Listing Module (M14)** | |  |
| **Function** | Description | Database |
| **M14F1** | **Content’s** listing view can be access by route **“/creator/content”** . In the view there will be a table having column **name, location**. All the respective data of **Content creator** will be shown from database.  Action column will be use to show action buttons such as  **Edit, Delete** etc. | List will use db table **“contentcreator”** and columns **id, title, image**, **pdf**, **video**, **description, crating, ccomplain**. |

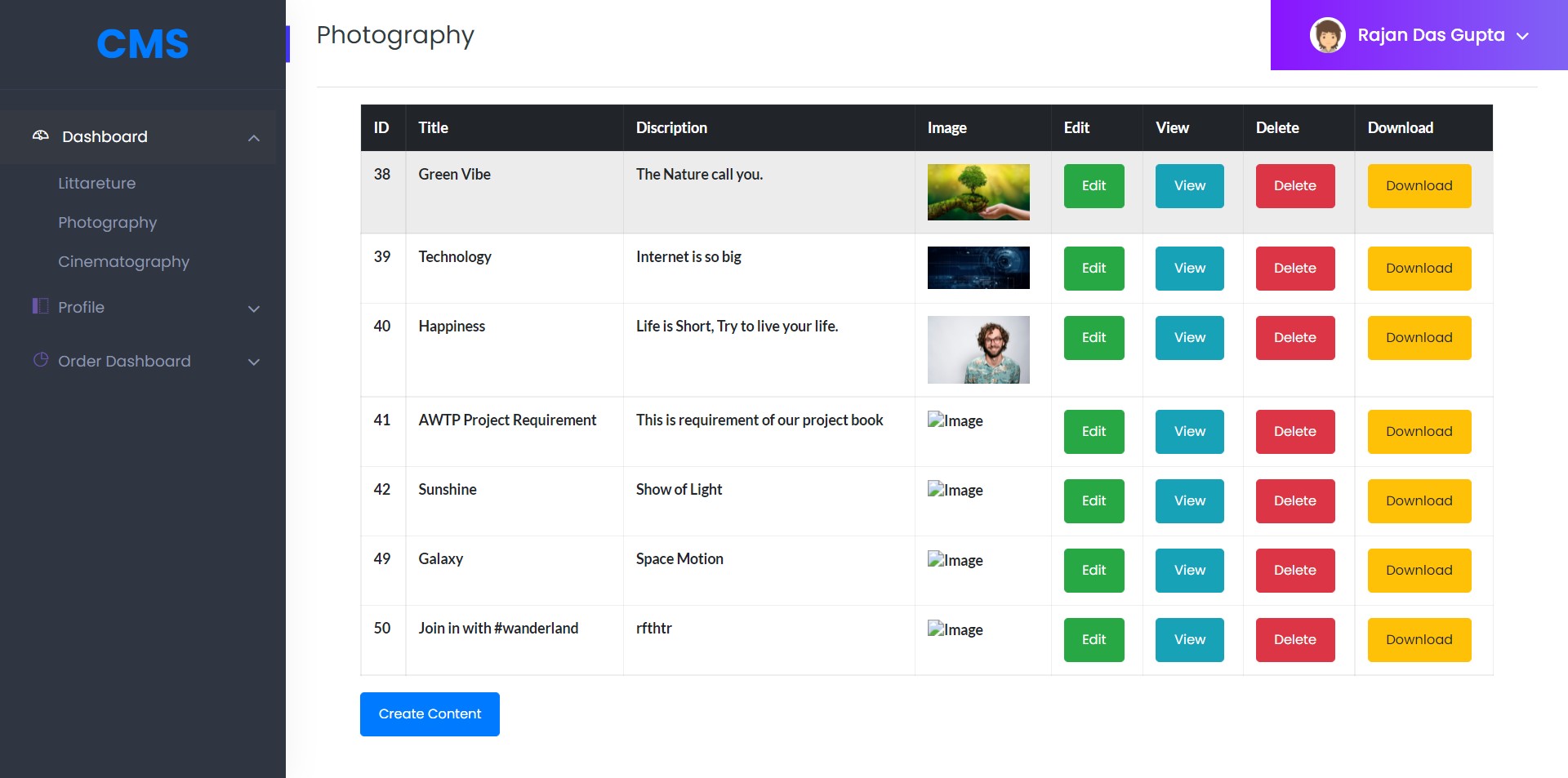
|  |  |  |
| --- | --- | --- |
| **Content Searching Module (M15)** | |  |
| **Function** | Description | Database |
| **M15F1** | It will use the view of **M13F1** to list the search result and can be access by same route. There’ll be a search field named **“search”** and a search submit button to fire the searching event**.** | List will use db table **“contentcreator”** and columns **title, image**, **pdf**, **video**, **description, crating, ccomplain**. |
| **M15F2** | Search event will call for an **Ajax (post) request** to ‘**creator/contents/ajax/search’** with the parameter **search** having value of search field. Once the search result is received then it will list on **M13F1 Listing** | Matches the search parameter with the db table **“contentcreator”** and columns **name, title, image**, **pdf**, **video.** Matching relation will be **OR** and pattern will be **LIKE**  **%%** |



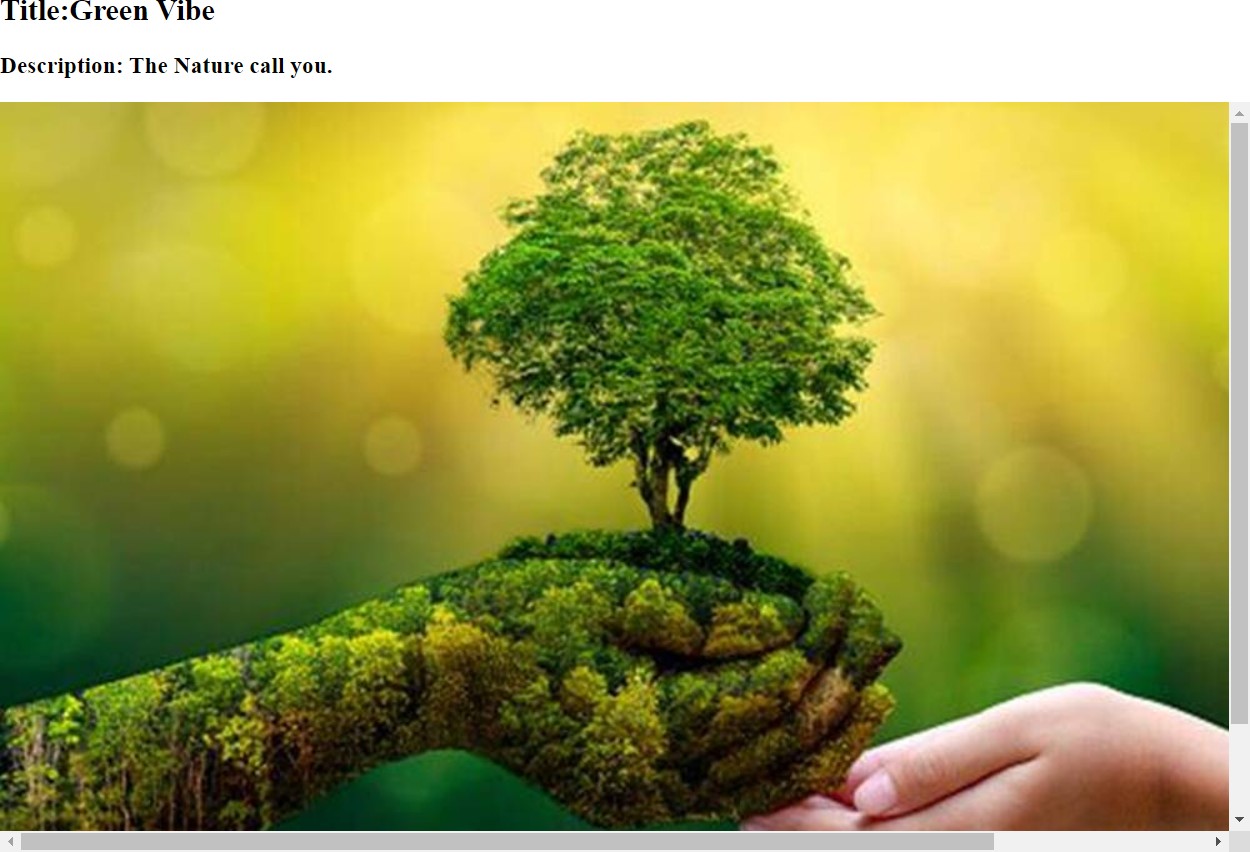
|  |  |  |
| --- | --- | --- |
| **Content Add (M16)** | |  |
| **Function** | Description | Database |
| **M16F1** | In the view of Module **M13F1** a Add New Button will be added referring the route **“/creator/content/add”** | N/A |
| **M16F2** | In the view there’ll be a html form having fields **name, email, password, password\_confirmation** and a submit button to fire form submission to the same route with post request. | N/A |
| **M16F3** | After receiving post request following form validation will be happen,   * title: Required * content: Required | Valid content| Unique * description: Required | Valid content| Unique   If everything is fine **title, image**, **pdf**, **video**, **description, crating, ccomplain** will be stored in the database. After then user will redirect to **M13F1** | Stores **title=title, image=image**, **pdf=pdf**, **video=video**, **description=description, crating=crating, ccomplain=ccomplain**.  to the db table **“contentcreator”** |



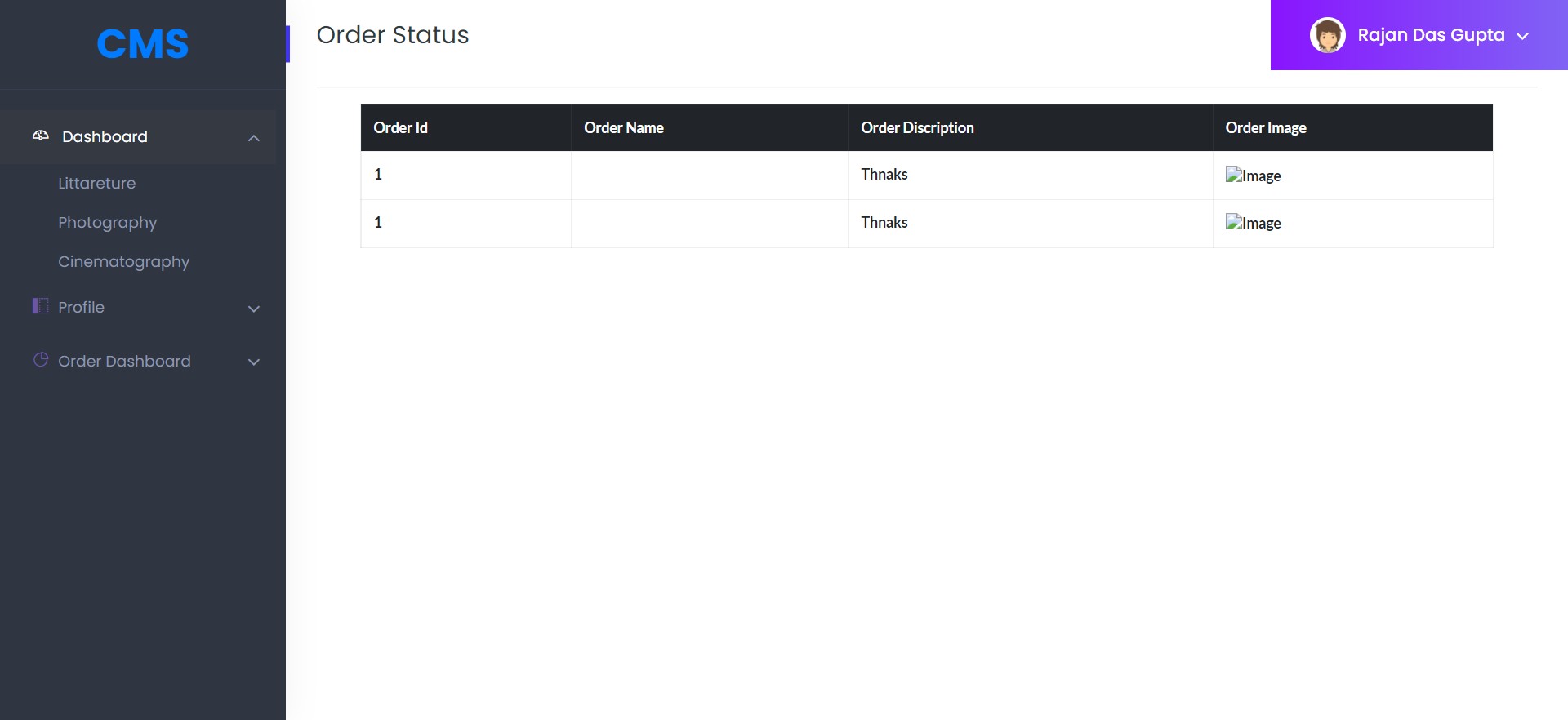
|  |  |  |
| --- | --- | --- |
| **Content Edit (M17)** | |  |
| **Function** | Description | Database |
| **M17F1** | In the List of Module **M13F1** a **Edit** Button will be added to every row referring the route **“/creator/content/{id}/edit”.** Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the table**“contentcreator”** |
| **M17F2** | In the view there’ll be a html form having fields **name, email, password** and a submit button to fire form submission to the same route with **Patch** request. | N/A |
| **M17F3** | After receiving post request following form validation will be happen,   * title: Required * content: Required | Valid content| Unique * description: Required | Valid content| Unique   If everything is fine **title, image**, **pdf**, **video**, **description, crating, ccomplain**will be stored in the database. After then user will redirect to **M13F1** | Find User from db table  **“contentcreator”** by **“id”**  Stores **title=title, image=image**, **pdf=pdf**, **video=video**, **description=description, crating=crating, ccomplain=ccomplain**.  to the db table **“contentcreator”** |



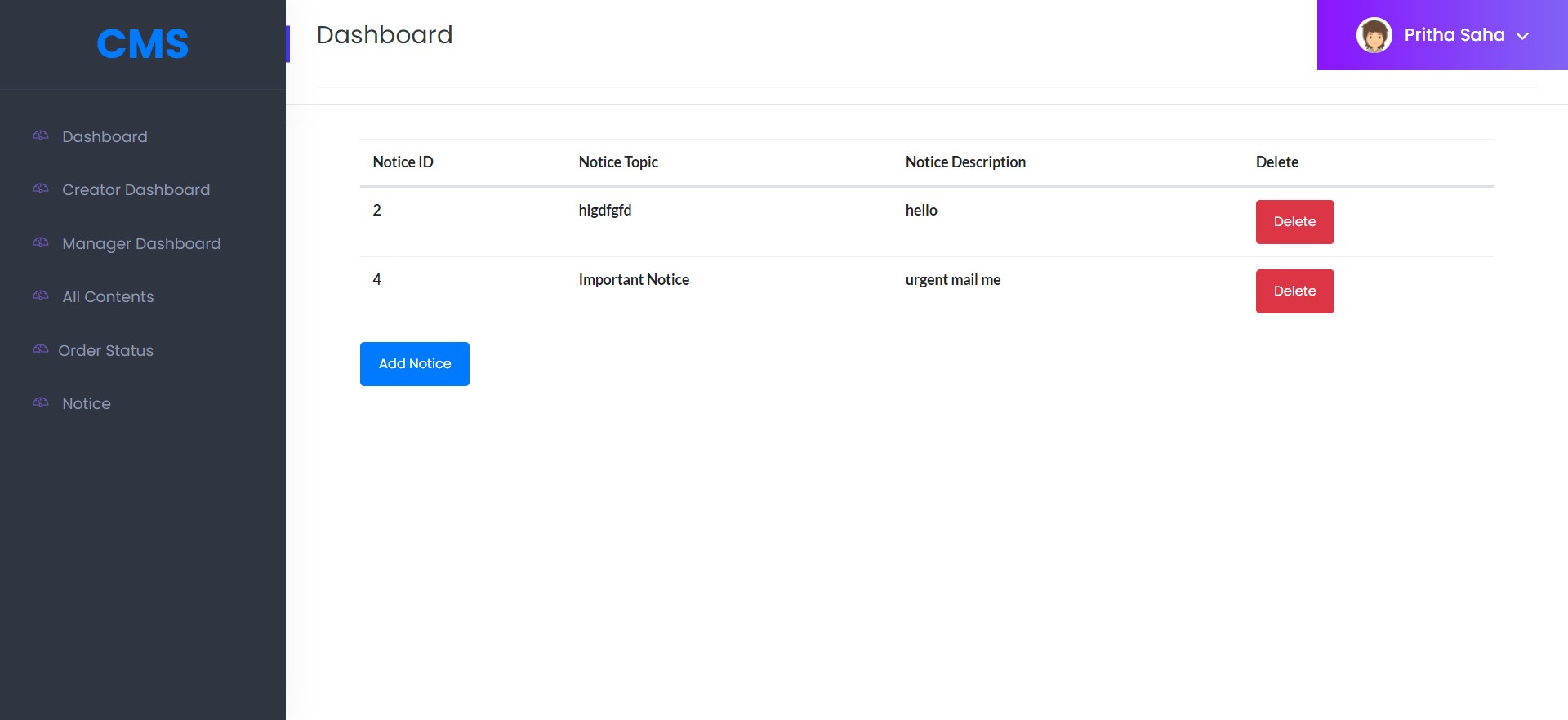
|  |  |  |
| --- | --- | --- |
| **Content Delete (M18)** | |  |
| **Function** | Description | Database |
| **M18F1** | In the List of Module **M13F1** a **Delete** Button will be added to every row referring the route **“/creator/content/{id}/delete”.** Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the table **“contentcreator”** |
| **M18F2** | In the view there’ll be a javascript confirmation**.** After confirm a virtual form will be submitted to the route of **M18F1** on **Delete** Request | N/A |
| **M18F3** | After receiving delete request support staff will be retrieved from the database and will be removed. After then user will redirect to **M13F1** | Find User from db table  **“contentcreator”** by **“id”**  Delete from db table  **“contentcreator”** by **“id”** |



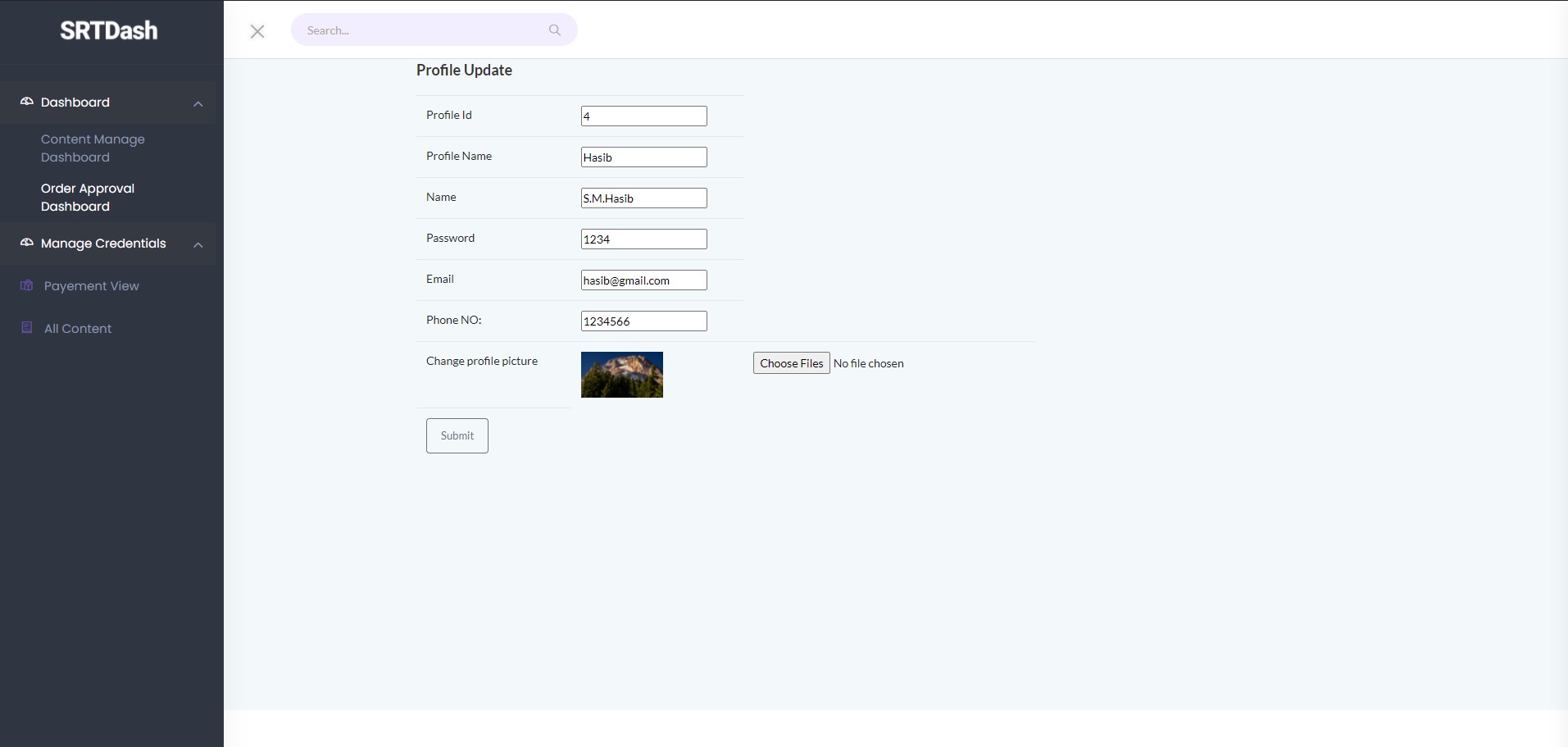
|  |  |  |
| --- | --- | --- |
| **Content Show - Creator(M19)** | |  |
| **Function** | Description | Database |
| **M19F1** | **Contents** listing view can be access by route **“/system/ contents”** . In the view there will be a table having column **title, image**, **pdf**, **video**, **description, crating, ccomplain\_column**. All the respective data will be shown from database. Action column will be use to show action buttons such as **Edit, Delete** etc. | List will use db table **“buses”** and **“contentcreator”** and columns **id, title, image**, **pdf**, **video**, **description, crating, ccomplain** |



|  |  |  |
| --- | --- | --- |
| **Content Orderlist Show - Creator(M19)** | |  |
| **Function** | Description | Database |
| **M19F1** | **Order List** view can be accessed by route **“/creator/ order/details”** by clicking **Order List** in **My orders** dropdown menu in side menu of dashboard In the view there will be a table having **Ordername, Order Description, Order Template, Order Image** and two buttons for **Edit** and **Delete.** List of all available content will be shown. | List will use db table **“orders”.** columns **ordername, orderdescription**, **template**, **picture**. |

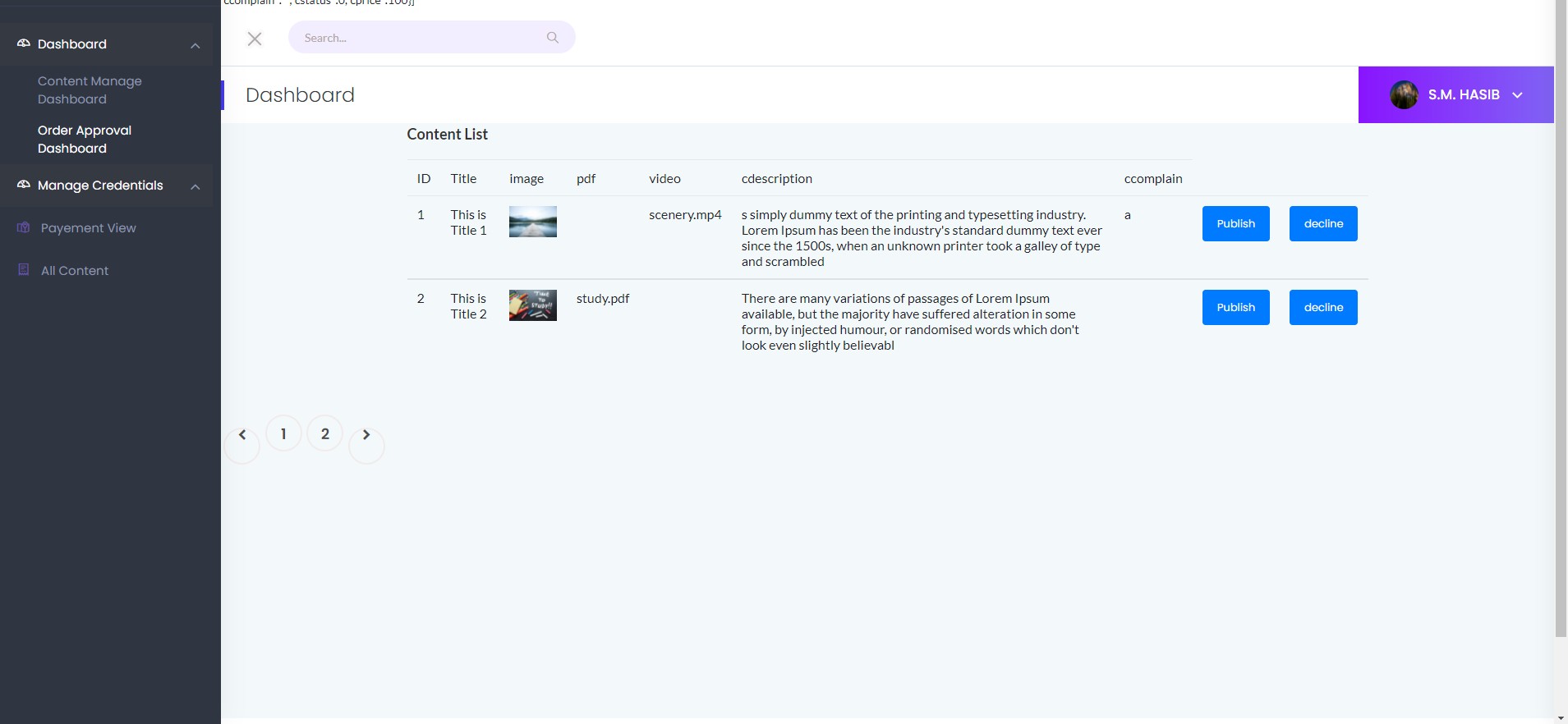


**Manager:**

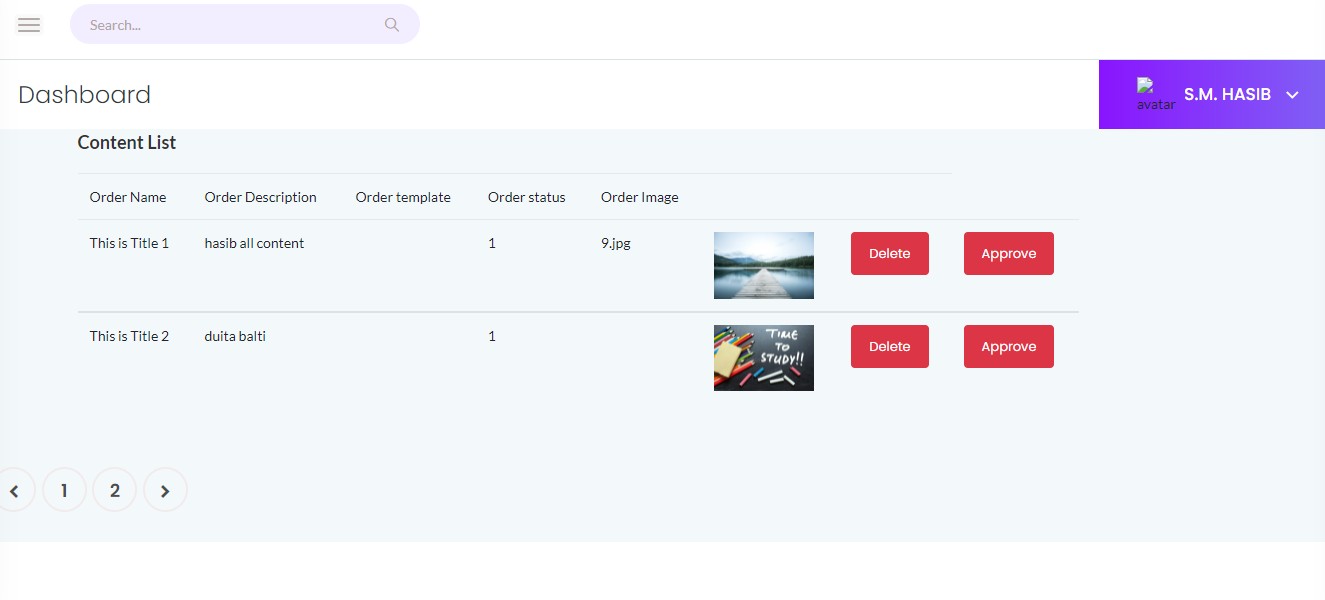


**Edit manager profile (M20)**

|  |  |  |
| --- | --- | --- |
| **Function** | Description | Database |
| **M20F1** | From Dashboard **Edit** Button will be added to every manager Dashboard referring the route **“/manager/editprofile/{id}”.** Here id parameter holds the id of manager | Uses the value of **“id”** column of the table**“manager”** |
| **M20F3** | In the view there’ll be a html form having fields **profileid , Profile name, email, password, phone no, profile image** and a submit button to fire form submission to the same route with **Post** request. | N/A |
| **M20F4** | After receiving **Post** request manager will be retrieved from the database and following form validation will be happen,   * Profile Name : Required * Name : Required * Email: Required | Valid Email | Unique except Current * Password: Nullable | Length:4 | Find User from db table **“users”** by **“id” ,** Updates **name=name, email=email, password=password,**  **type=type** to the db table  **“users”** |
| **M20F5** | after clicking submit button all the data from the form will be stored to the database | update database by  id,name,username,password,email,p phoneno, profile pic |

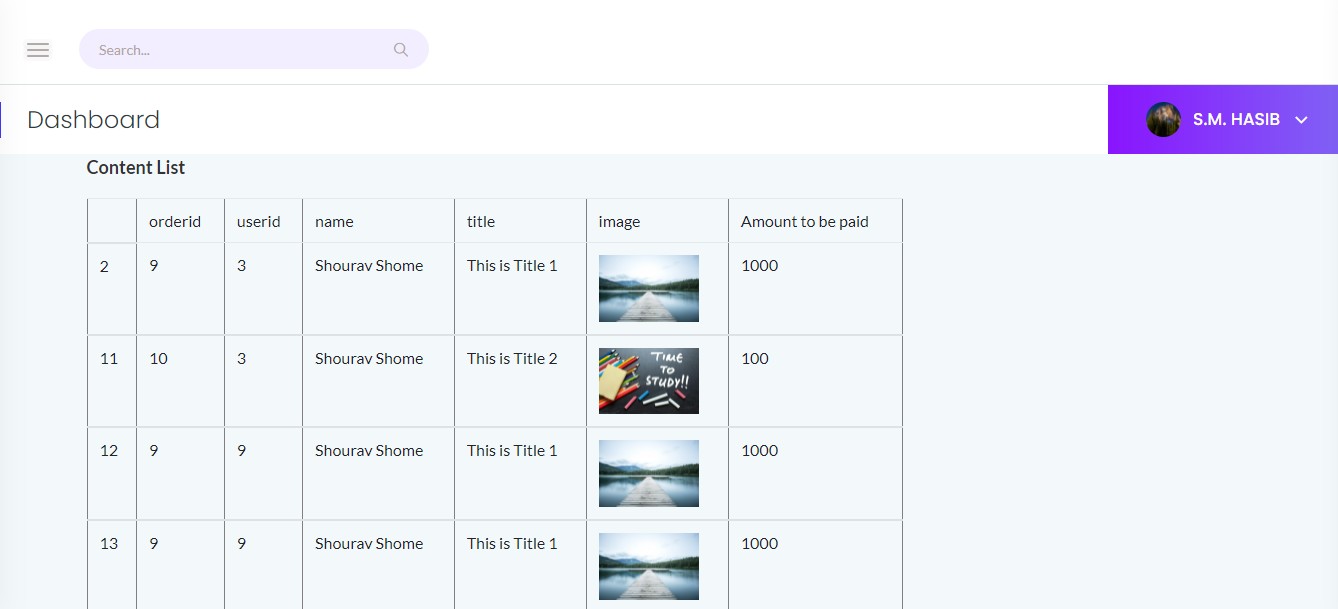


|  |  |  |
| --- | --- | --- |
| **Contest publish approve module** | |  |
| **Function** | Description | Database |
| **M21F1** | **content request** listing view can be access by route **“/user/ dashboard”** . In the view there will be a table having column **id, title, image, pdf, video**. **description, complain.** All the respective data of **requests** will be shown from database. Action column will be use to show action buttons such as **Publish , Decline** etc. | List will use db table **“content creator”** and columns **id, title, image, cstatus, description**  **,picture** |
| **M21F2** | By clicking **publish** button , a post request will pass through **contentpublish/{id}** to pass the corresponding id to change cstatus in database | update column **cstatus 1** in table  **content creator** |
| **M20f3** | By clicking **decline** button , a post request will pass through **contentpublish/{id}** to pass the corresponding id to delete the row in database table | delete the column of corresponding id of the row |

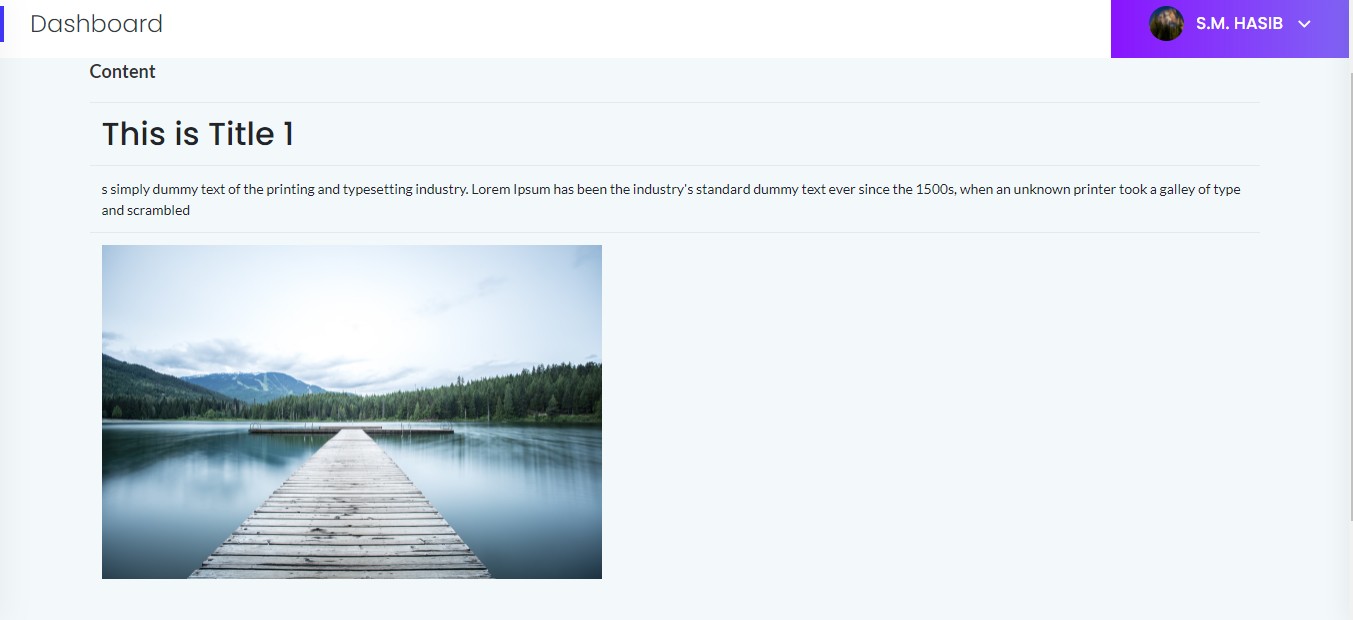


|  |  |  |
| --- | --- | --- |
| **Orer approve Module (M21)** | |  |
| **Function** | Description | Database |
| **M21F1** | **order** listing view can be access by route **“/user/ order/MNGdetails”** . In the view there will be a table having column **ordername, order description, order template, orderstatus, orderimage .** All the respective data of **requests** will be shown from database. Action | List will use db table **“order”** and columns **id, ordername,order status, image.** |

|  |  |  |
| --- | --- | --- |
|  | column will be use to show action buttons such as  **Delete, approve** etc. |  |
| **M21F2** | **approve** will generate a post request to **user/order/approve/{id}** to update a value of **orderstatus** and will list on **M21 F1 Listing** | **update orderstatus** column from  **order** table 0/1 |
| **M21F3** | **delete** will generate a post request to **user/order/delete/{id}** to update a value of **orderstatus** and will list on **M21 F1 Listing** | **delete** a row from **order** table corresponding to the **id** |



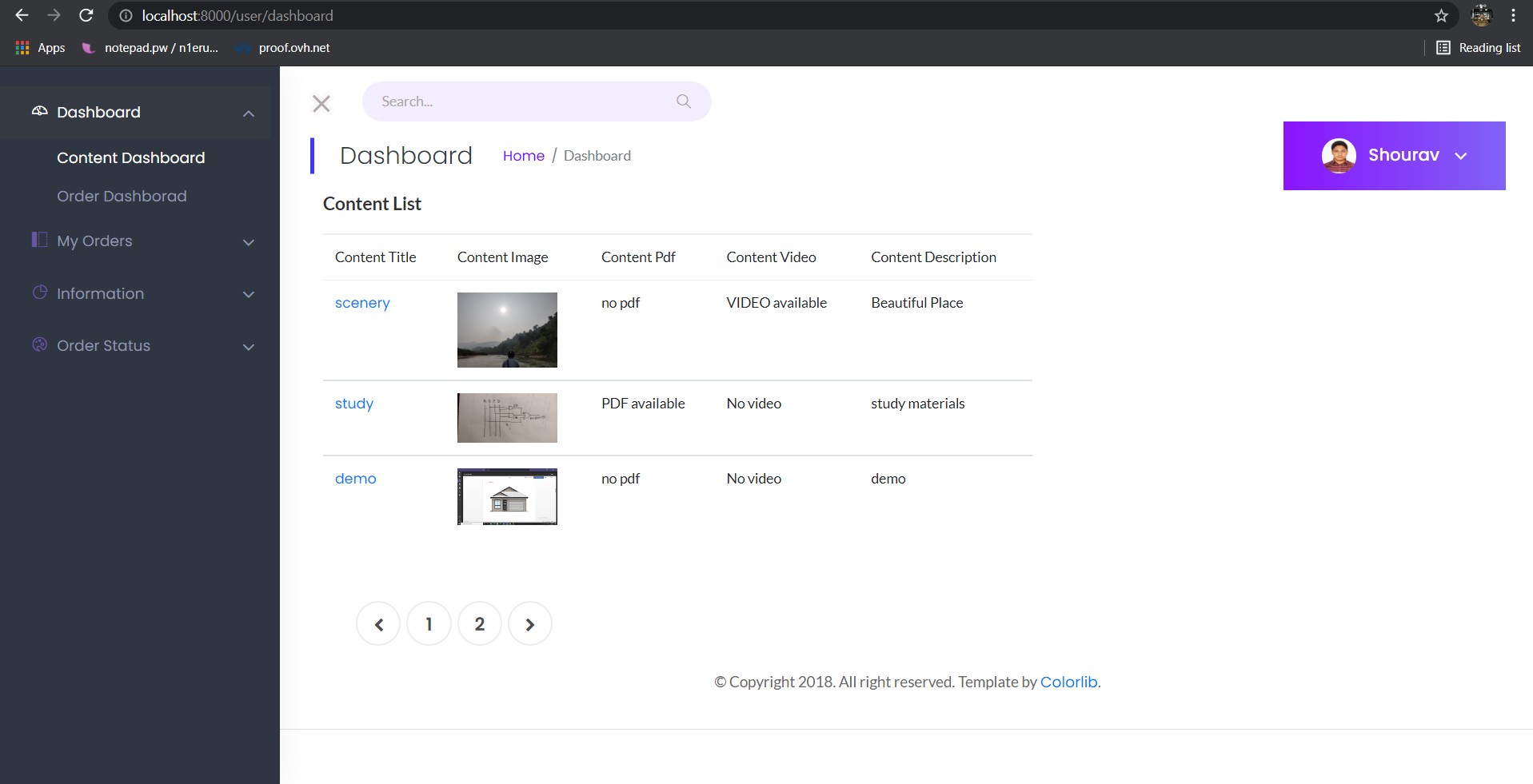
|  |  |  |
| --- | --- | --- |
| **Payment View module(M22)** | |  |
| **Function** | Description | Database |
| **M22F1** | **payment list request** listing view can be access by route **“/user/payment”** . In the view there will be a table having column p**id, uid, orderid, name, title, image, amount.** | List will use db table **“payment”** and columns p**id, uid, orderid, name, title, image, amount.** |



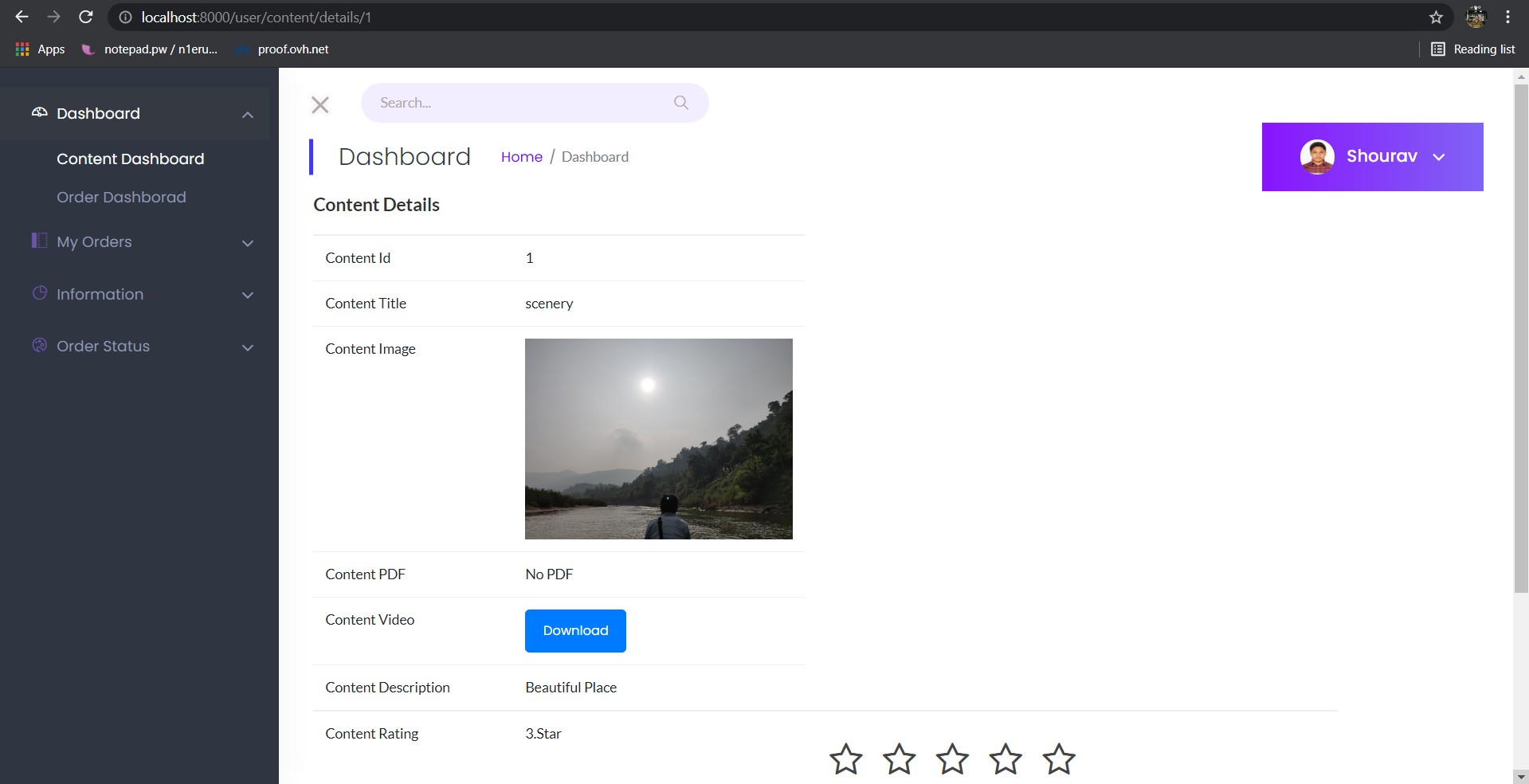
|  |  |  |
| --- | --- | --- |
| **Published content View module(M23)** | |  |
| **Function** | Description | Database |
| **M23F1** | **payment list request** listing view can be access by route **“/user/allcontent”** . Title of the content , Picture of the content and Descripiton, author name | List will use db table **“payment”** and columns p**id, uid, orderid, name, title, image, amount.** |

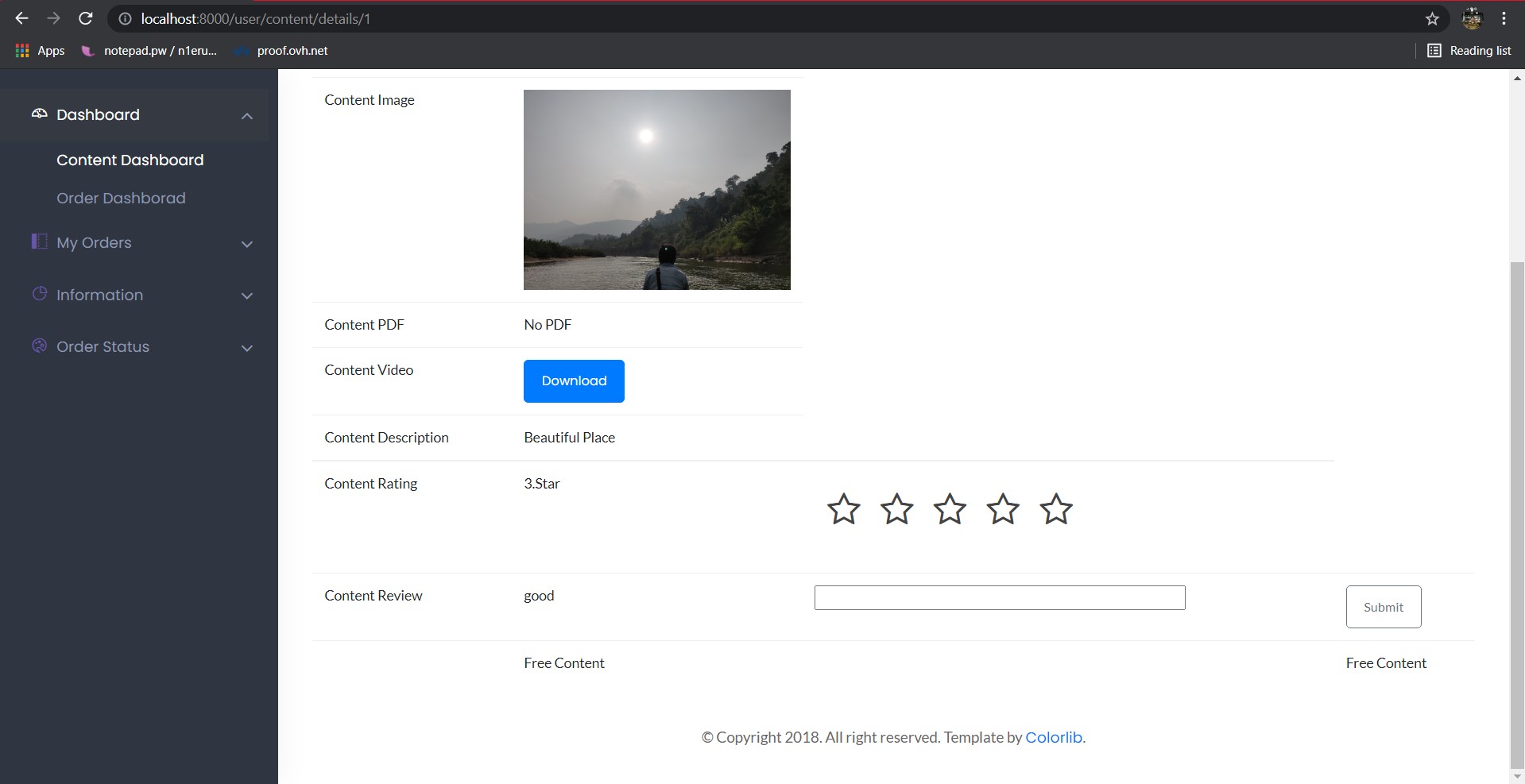
Access control:

|  |  |
| --- | --- |
| **Role name** | **Module(M),Functions(F)** |
| Admin | * M1F1,M1F2,M1F3,M1F4 * M2F1 * M4F1 * M5F1,M5F2 * M6F1,M6F2,M6F3 * M7F1,M7F2,M7F3 * M8F1 * M9F1,M9F2 * M10F1,M10F2,M10F3 * M11F1,M11F2,M11F3 * M12F1,M12F2,M12F3 * M13F1 * M15F2,M15F2 * M16F1,M16F2,M16F3 * M17F1,M17F2,M17F3 * M18F1,M18F2,M18F3 * M19F1 * M21F1,M21F2 * M22F1,M22F2,M22F3 * M23F1,M23F2,M23F3 * M24F1,M24F2,M24F3 * M25F1 * M27F1,M27F2 * M28F1,M28F2,M28F3 * M29F1,M29F2,M29F3 * M30F1,M30F2,M30F3 * M31F1 * M32F1,M32F2,M32F3 * M33F1 |

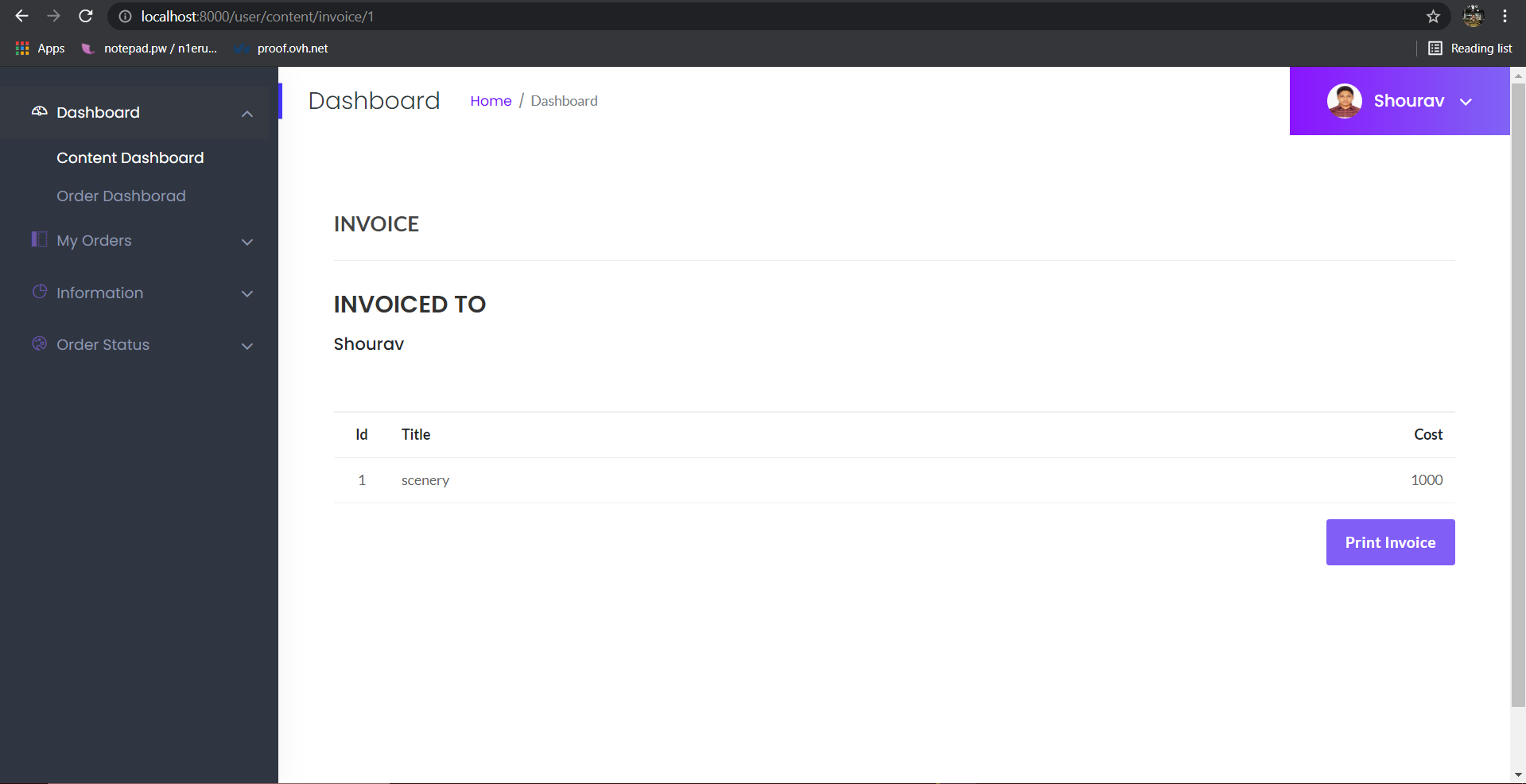
USER:

|  |  |  |
| --- | --- | --- |
| **Content List Module** | |  |
| **Function** | Description | Database |
|  | **Content List** view can be accessed by route **“/user/** | List will use db table |
| **dashboard”** . In the view there will be a table having | **“contentcreator”.** columns **id,** |
| **Content Title, Content Image, Content Pdf, Content** | **title, image**, **pdf**, **video**, |
| **Video, Content Description**. List of all available content | **description**. |
| will be shown. |  |

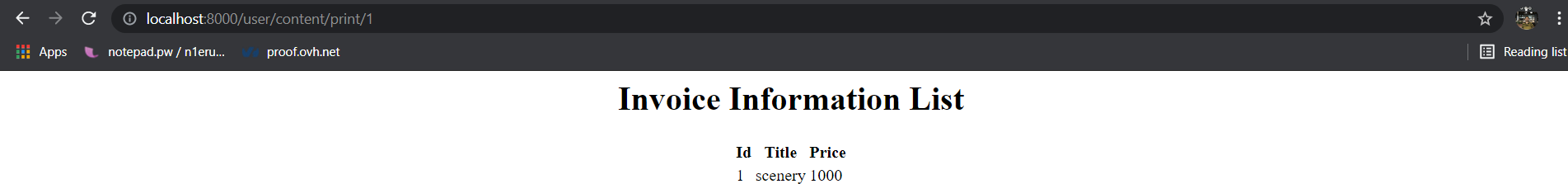




|  |  |  |
| --- | --- | --- |
| **Single Content Details View, Rating and Review Submission**  **Module** | |  |
| **Function** | Description | Database |
|  | **Single Content Details** can be accessed by clicking on title of a content in **Content List.** Route for Details is **“/user/content/details/{id}”** . In the view there will be a table having **Content Title, Content Image, Content Pdf, Content Video, Content Description, Content Rating and Content Review**. And two buttons For **Buy** and **Add to wishlist** option. And a submit button to fire form submission to the same route for Rating and Review. | Find content from **“contentcreator” table** using “**id**”**.** columns **id, title, image**, **pdf**, **video**, **description, crating, ccomplain**. |
|  | In the view There is a row for **Content Rating** and **Content Review.** There are Five stars in this row to give a rating from 1 to 5. each star represents a star and is stored in database as numerical value. And a text field to write review of content in the row **Content Review** which is also stored in database. | Find ontent from db table  **“contentcreator”**  by **“id”**  Updates **crating=star, ccomplain=complain,** to the db table **“contentcreator”** |

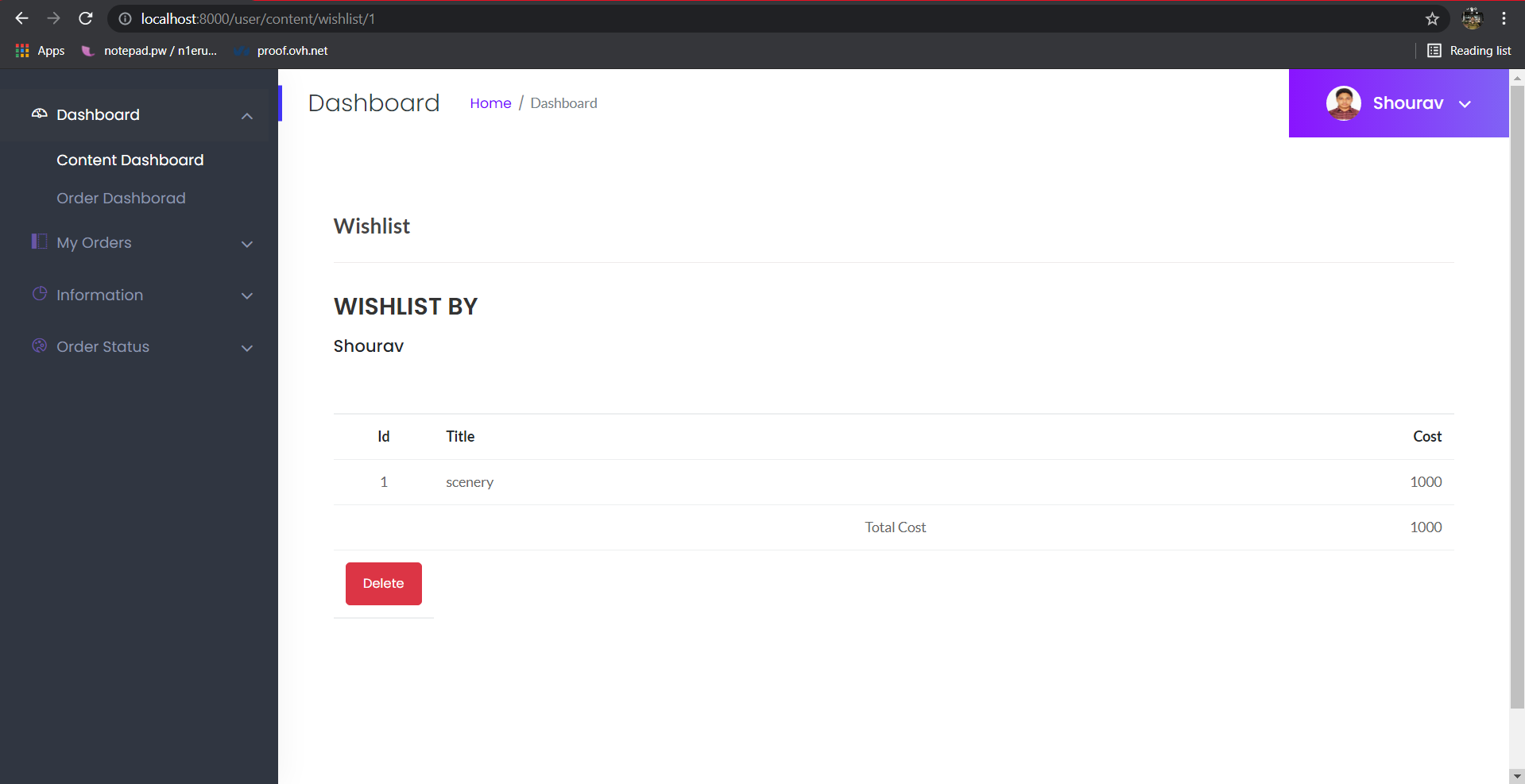


|  |  |  |
| --- | --- | --- |
| **Buy Content or Invoice Module(M11)** | |  |
| **Function** | Description | Database |
|  | In the List of Module **Single Content Details view** a **Buy** Button will be added referring the route **“/user/content/invoice/{id}”.** Here id parameter holds the content id. | Uses the value of **“id”** column of the db table **“users”** |
|  | In the view there’ll be a html form having fields **content id, Title, Cost** and a button to print the invoice list. | Find content from db table  **“contentcreator”**  by **“id”**  columns “**id**”, “**title**”, “**cprice**”. |



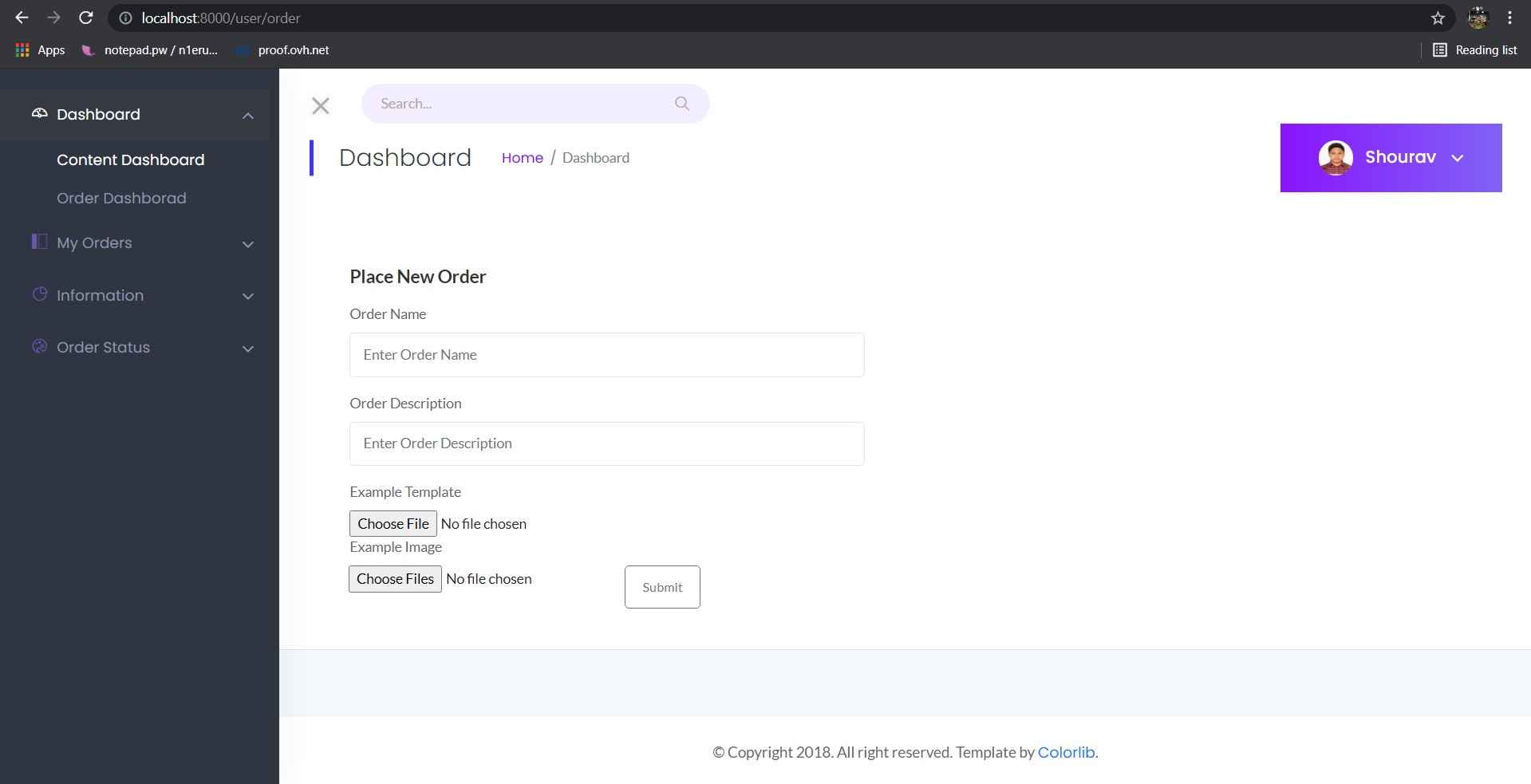
|  |  |  |
| --- | --- | --- |
| **Print Invoice** | |  |
| **Function** | Description | Database |

|  |  |  |
| --- | --- | --- |
|  | In the Invoice Module there is a Button referring the route **“/user/content/print/{id}}”.** Here id parameter holds the content id. | N/A |
|  | In the view there’ll be a html form having fields  **content id, Title, Price.** | N/A |

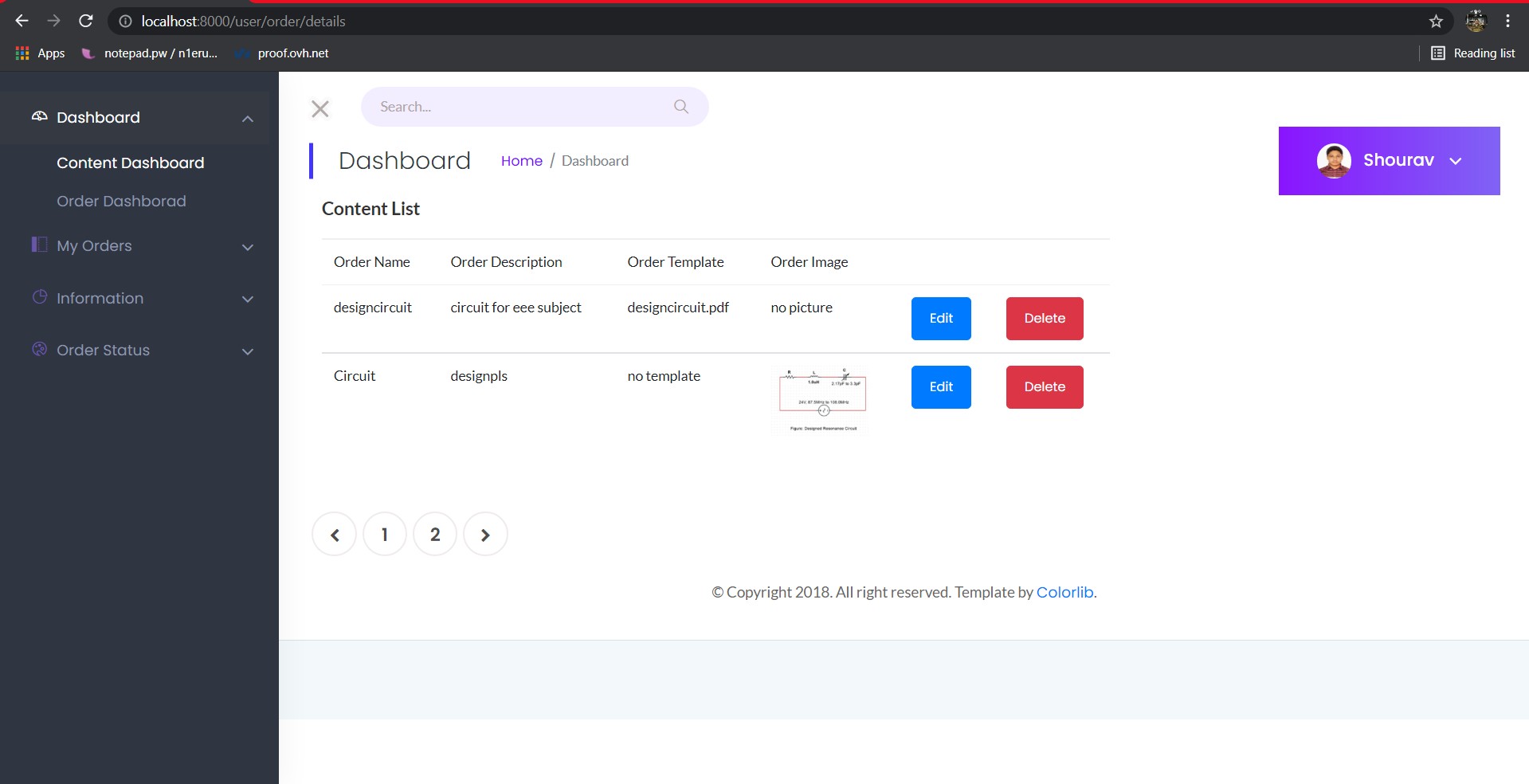


|  |  |  |
| --- | --- | --- |
| **Wishlist Module View or Delete(M11)** | |  |
| **Function** | Description | Database |
|  | In the List of Module **Single Content Details view** an **Add to wishlist** Button will be added referring the route **“/user/order/wishlist/{id}”.** Which can also be accessed from Dashboard under Order Status dropdown menu. Here id parameter holds the content id. | Find content from db table  **“contentcreator”**  by **“id”** select value of columns “**id**”, “**title**”, “**cprice**”, “**wstatus**”. and store in db table “**wishlist**” columns  “cid”=id “ctitle”=title “cstatus”=wstatus  “cprice”=cprice |
|  | In the view there’ll be a html form having fields **content id, Title, Cost, Total Cost** which is calculated by price of all content available in wishlist and a button to **Delete** the wishlist. | Find content from db table  **“contentcreator”**  by **“wstatus” if wstatus=1 then select columns** “**id**”, “**title**”, “**cprice**”. |
|  | If **Delete** button is clicked same html form is requested but the wishlist is cleared. | truncate all content from db table “**wishlist**” and change “**wstatus**” of all rows to 0 in db tabel |

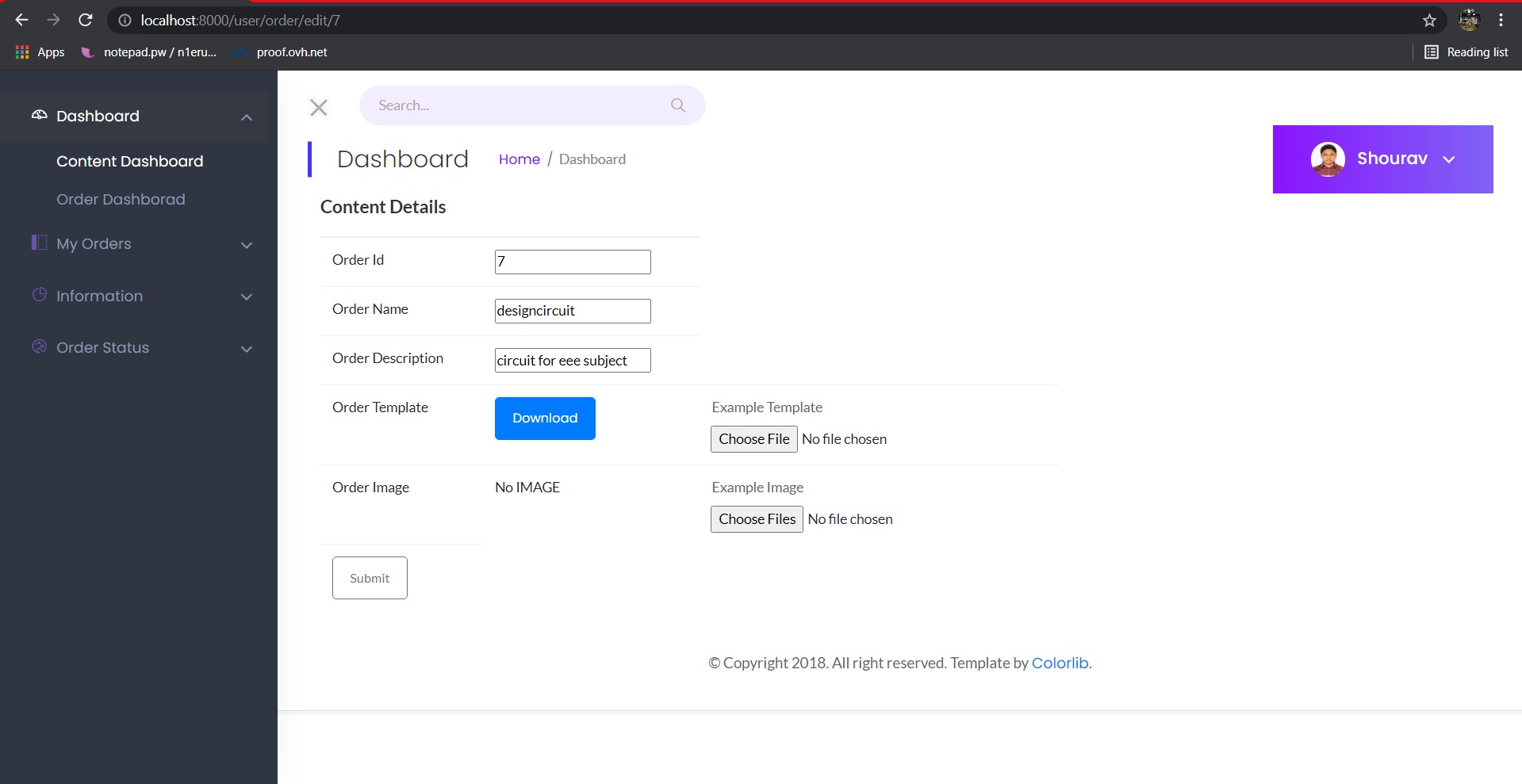
|  |  |  |
| --- | --- | --- |
|  |  | “**contentcreator**” |



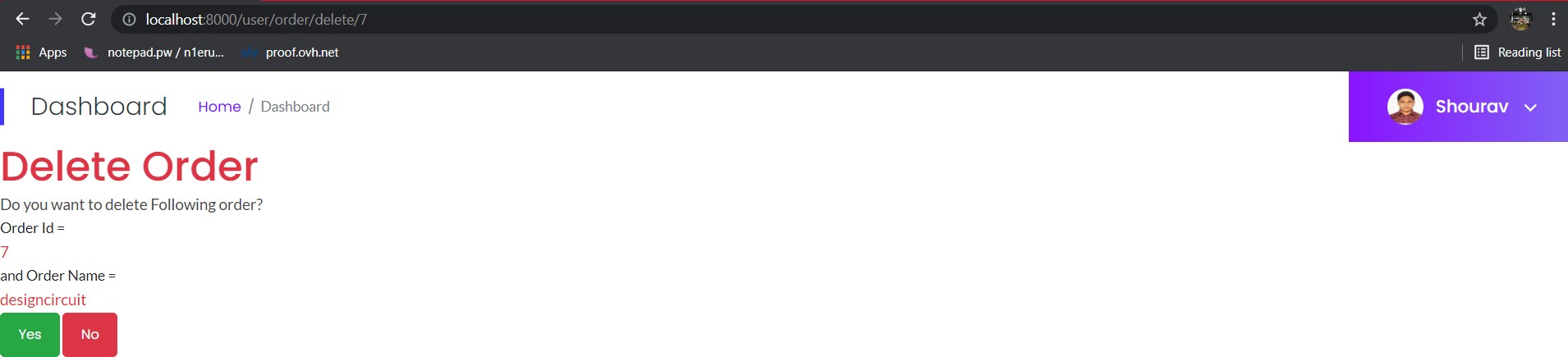
|  |  |  |
| --- | --- | --- |
| **Order add Module** | |  |
| **Function** | Description | Database |
|  | In the sidemenu of dashboard a order dashboard button is present in dropdown menu referring the route “**/user/order**” | N/A |
|  | In the view there’ll be a html form having fields **place new order, order name, Order description, example template, example image** and a submit button to fire form submission to the same route with post request. | N/A |
|  | After receiving post request following form validation will be happen,   * Order Name: Required * Order Description: Required |   If everything is fine **order name, order description** will be stored in the database. After then user will redirect to | Stores **ordername=ordername, orderdescription=orderdescri ption, template=temp, picture=pic,** to the db table **“orders”** |



|  |  |  |
| --- | --- | --- |
| **Order List Module** | |  |
| **Function** | Description | Database |
|  | **Order List** view can be accessed by route **“/user/ order/details”** by clicking **Order List** in **My orders** dropdown menu in sidemenu of dashboard In the view there will be a table having **Ordername, Order Description, Order Template, Order Image** and two buttons for **Edit** and **Delete.** List of all available content will be shown. | List will use db table **“orders”.** columns **ordername, orderdescription**, **template**, **picture**. |

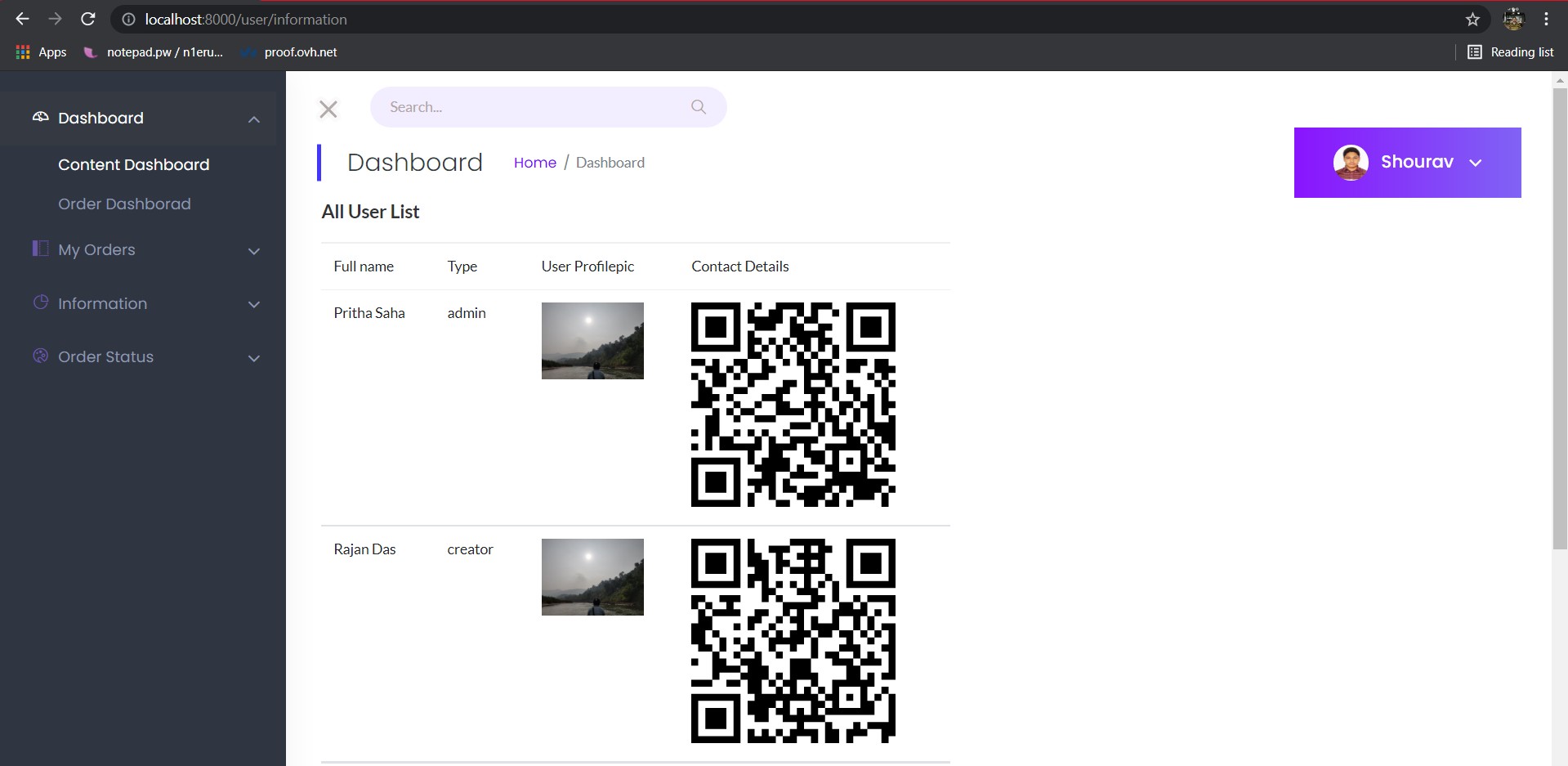


|  |  |  |
| --- | --- | --- |
| **Order Edit** | |  |
| **Function** | Description | Database |
|  | In the Order List Module a **Edit** Button will be added to every row referring the route **“/user/order/edit/{id}”.** Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“orders”** |
|  | In the view there’ll be a html form having fields **order id, order name, order description, order template, order image** and a submit button to fire form submission to the same route with **Patch** request. | N/A |
|  | After receiving patch request support staff will be retrieved from the database and following form validation will be happen,   * Order Name : Required   If everything is fine **Order Name** will be saved in the database. | Find User from db table **“orders”**  by **“id”**  Updates **ordername= ordername, orderdescription=orderdescr iption, template=temp, picture=pic** to the db table **“orders”** |

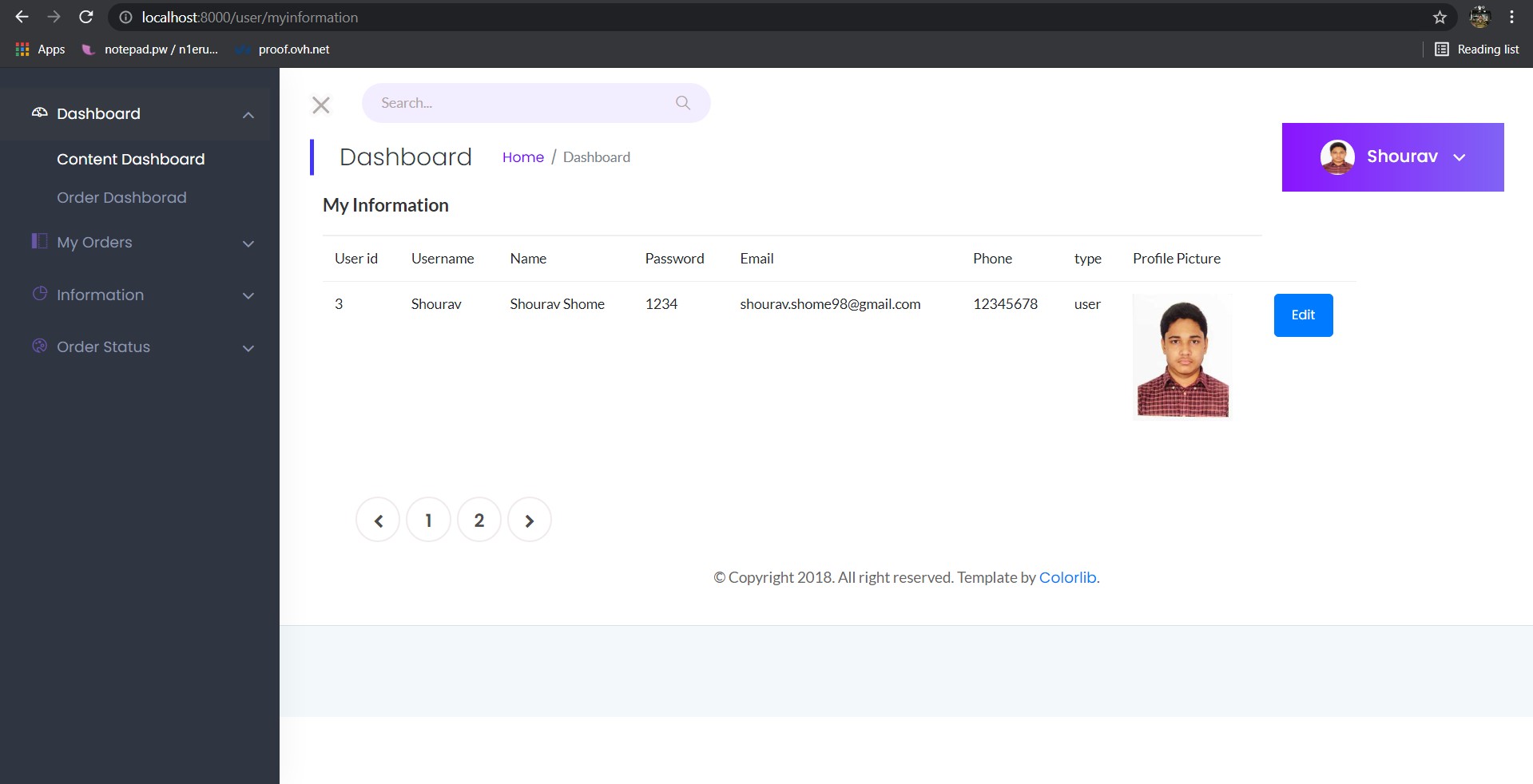


|  |  |  |
| --- | --- | --- |
| **Order Delete** | |  |
| **Function** | Description | Database |
| **M24F1** | In the Order List Module a **Delete** Button will be added to every row referring the route **“/user/order/delete/{id}”.** Here id parameter holds the staff id of a row. | Uses the value of **“orderid”** column of the db table **“orders”** |
| **M24F2** | In the view there’ll be a html page asking for confirmation**.** After confirm a virtual form will be submitted on **Delete** Request | N/A |

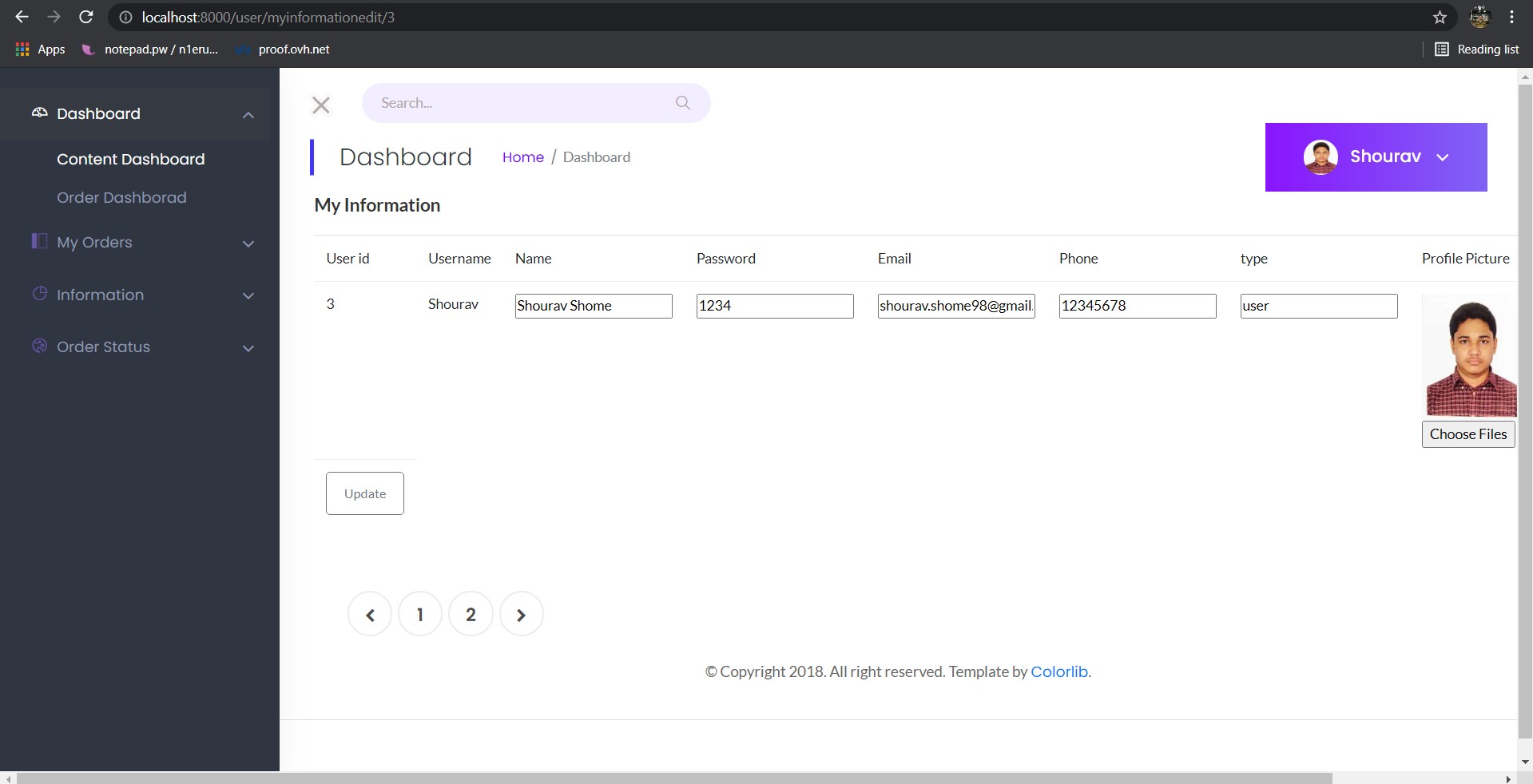
|  |  |  |
| --- | --- | --- |
| **M24F3** | After receiving delete request order list will be retrieved from the database and will be removed. After then user will redirect to **Order List Modult** | Find Order from db table **“orders”**  by **“orderid”**  Delete from db table **“orders”** by  **“orderid”** |



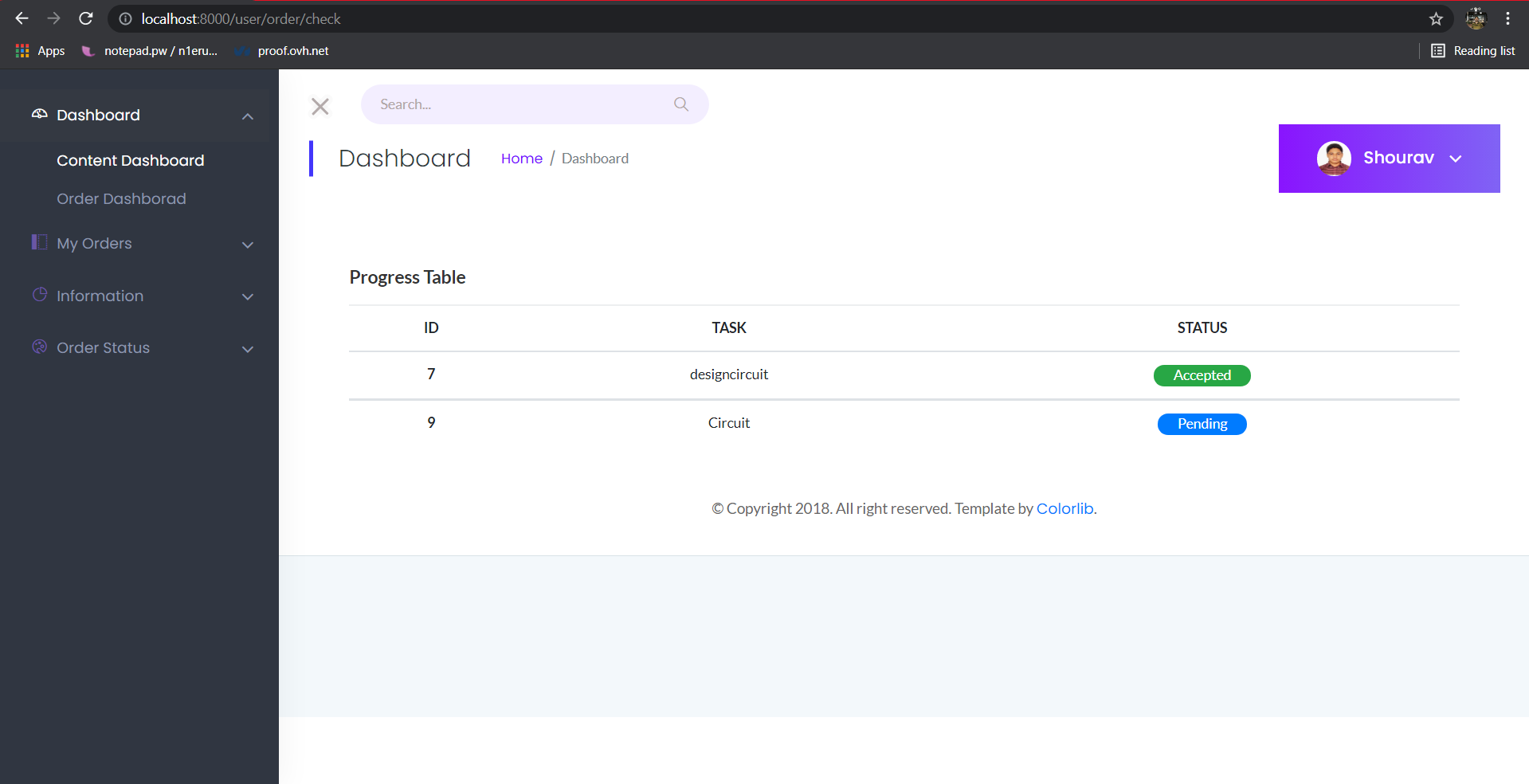
|  |  |  |
| --- | --- | --- |
| **All User List Module** | |  |
| **Function** | Description | Database |
|  | **User List** view can be accessed by route **“/user/ information”** by clicking **User Information** in **Information** dropdown menu in sidemenu of dashboard In the view there will be a table having **Full name, Type, User Profilepic, Contact Details.** Contact details columbine has a **QR code** for every row holding **Phone no** and **Email** of the user.. | List will use db table **“users”.** columns **name, type**, **profilepic**, **email, phoneno**. |



|  |  |  |
| --- | --- | --- |
| **Current User View Module** | |  |
| **Function** | Description | Database |
|  | **Current User List** view can be accessed by route **“/user/ myinformation”** by clicking **My Information** in **Information** dropdown menu in sidemenu of dashboard In the view there will be a table having **User id, Username, Name, Password, Email, Phone, type, Profile Picture** and a **Edit** button**.** | Find User from db table **“users”**  by **“username”**  List will use db table **“users”.** columns **userid, username, name, password, type**, **profilepic**, **email, phoneno**. |



|  |  |  |
| --- | --- | --- |
| **Current User Edit Module** | |  |
| **Function** | Description | Database |
|  | In the Current User View Module a **Edit** Button will be added to teh row referring the route **“/user/myinformationedit/{id}”.** Here id parameter holds the staff id of a row. | Uses the value of **“id”** column of the db table **“users”** |
|  | In the view there’ll be a html form having fields **User id, Username, Name, Password, Email, Phone, type, Profile Picture** and a submit button to fire form submission to the same route with **Patch** request. | N/A |
|  | After receiving patch request support staff will be retrieved from the database and following form validation will be happen,   * Name : Required * Password : Required * Email : Required * Phone : Required * type : Required   If everything is fine **User** will be updated in the database. | Find User from db table **“users”**  by **“userid”**  Updates **name=name, password=password, email=email, phoneno=phoneno, type=type, profilepic=picture** to the db table **“users”** |



|  |  |  |
| --- | --- | --- |
| **Order Status Check Module** | |  |
| **Function** | Description | Database |
|  | **Order Status Check** view can be accessed by route **“/user/ order/check”** by clicking **Check** in **Order Status** dropdown menu in sidemenu of dashboard. In the view there will be a table having **ID, TASK, STATUS.** In status column there is two option one is **Accepted** and other is **Pending.** | Find Order from db table **“orders”**  by **“orderstatus”**  List will use db table **“orders”.** columns **orderid, ordername, orderstatus.** |