OUTPUT SCREEN OF PROJECT

Project Report

On

Online Notes Sharing System

REPORT INDEX

ACKNOWLEDGEMENT

I am are over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

I would like to express my special thanks of gratitude to my esteemed guide, Mr. XYZ who gave me the golden opportunity to do this wonderful project which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them.
Any attempt at any level can't be satisfactorily completed without the support and guidance of MY parents and friends.

I would like to thank my parents who helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project, despite of their busy schedules, they gave me different ideas in making this project unique.

.

ABSTRACT

“Online Notes Sharing System” is a web based technology which manages user and notes database and provides rights to update their details and notes.

This project manage user information and notes which is provided by him/her. User’s also have rights to update their details and notes. In this web application user must be registered. The aim of this web application is to manage user and notes database.

INTRODUCTION

“Online Notes Sharing System” is a web based technology which manages user and notes database and provides rights to update their details and notes. In this web application user must be registered. This web application provides a way to effectively control record & track the user details who himself/herself registered with us.

Online Notes Sharing System provides functionality to manage personal profiles and manages notes which are provided by them. Personal profiles contain information such as profile contact number, email, first and last name, etc. and notes details contains notes title, notes subject , notes descriptions etc. In order to access personal profile for reading and modification password is used for authentication. User information contains username, password and session for authentication.

In this project we use Python, Django Framework and MySQL database and it has one module i.e. **User**

User:

1. **Dashboard:** This is welcome page of registered users, user can briefly view total number of notes uploaded and files uploaded.
2. **Notes:** In this section, user can add, update and delete the notes which they want to share.
3. **My Accounts:** This section have following:
   - **Profile:** User can update his/her profile.
   - **Change Password:** User can update/his/her account password.
   - **Logout:** User can logout from their panel.

4. **Search:** User can search their notes by the titles and subject.

Advantages:

- It helps to handle and manage user data and their notes.
- Reduce time consumption.
- Reduce error scope.
- All system managements are automated.
- Centralized database management.
- Easy operations for operator of the system.
- No paper work requirement.

Disadvantages:

- The system can only handle single user.

FEASIBILITY STUDY

Whenever we design a new system, normally the management will ask for a feasibility report of the new system. The management wants to know the technicalities and cost involved in creation of new system.

- Technical feasibility
- Economic feasibility
- Physical feasibility

Technical feasibility:

Technical feasibility involves study to establish the technical capability of the system being created to accomplish all requirements to the user. The system should be capable of handling the proposed volume of data and provide users and operating environment to increase their efficiency.
For example, system should be capable of handling the proposed volume of data and provide users.

Economic feasibility:

Economic feasibility involves study to establish the cost benefit analysis. Money spent on the system must be recorded in the form of benefit from the system. The benefits are of two types:

Tangible benefits:

- Saving man labor to do tedious tasks saves time.
-

Intangible benefits:

- Improves the quality of organization.

Physical feasibility:

It involves study to establish the time responses of the new system being created. For e.g., if the new system takes more than one day to prepare crucial finance statement for the management, wherever it was required in an hour, the system fails to provide the same.

It should be clearly establish that the new system requirements in the form of time responses would be completely met with. It may call for increase in cost. If the required cost is sacrificed then the purpose of the new system may not be achieved even if it was found to be technically feasible.

SCOPE OF THE PROJECT

The proposed system provides a simple user and notes management pages that lets you view users and notes details, create new user accounts, edit existing ones and manage notes which is share by them. Also provides tools to disabled or delete users, manage permission and issue password reset.

The system works and fulfills all the functionalities as per the proposed system.

It will provide reduced response time against the queries made by different users.

All possible features such as verification, validation, security, user friendliness etc have been considered.

SOFTWARE AND HARDWARE REQUIREMENTS

- ✓ Any Version of browser after Mozilla Firefox 4.0, Internet Explorer 6.0,chrome

Hardware requirements:

- ✓ Any processor after Pentium 4.
- ✓ Any version of Windows XP or later.
- ✓ Processor speed: 2.0 GHz
- ✓ RAM : 1GB
- ✓ Hard disk: 40GB to 80 GB

Software requirements:

- ✓ Database : MySQL
- ✓ Server : Apache
- ✓ Frontend : HTML
- ✓ Scripting Language : JavaScript
- ✓ IDE : PyCharm
- ✓ Technology : Python
- ✓ Framework : Django

SYSTEM DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data

Unified Modelling Language Diagrams (UML):

- The unified modelling language allows the software engineer to express an analysis model using the modelling 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 different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

i. This view represents the system from the users perspective.

ii. The analysis representation describes a usage scenario from the end-users perspective**.**

Structural model view

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

Behavioural Model View

- It represents the dynamic of behavioural 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 behavioural as parts of the system are represented as they are to be built.

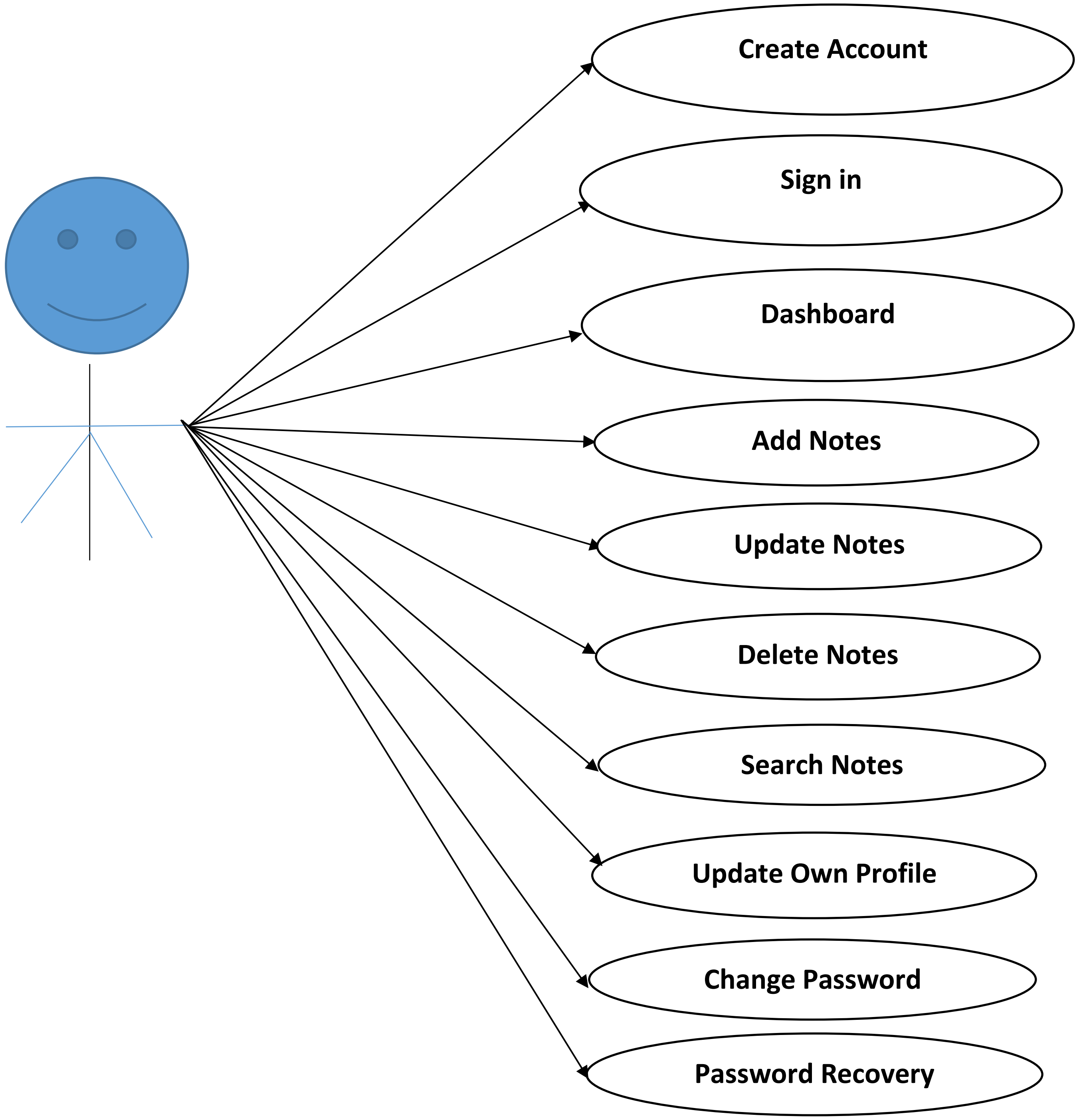
Environmental Model View

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

UML is specifically constructed through two different domains they are

- UML Analysis modelling, which focuses on the user model and structural model views of the system?
- UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

Use Case Diagrams User



ENTITY-RELATIONSHIP Diagrams

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

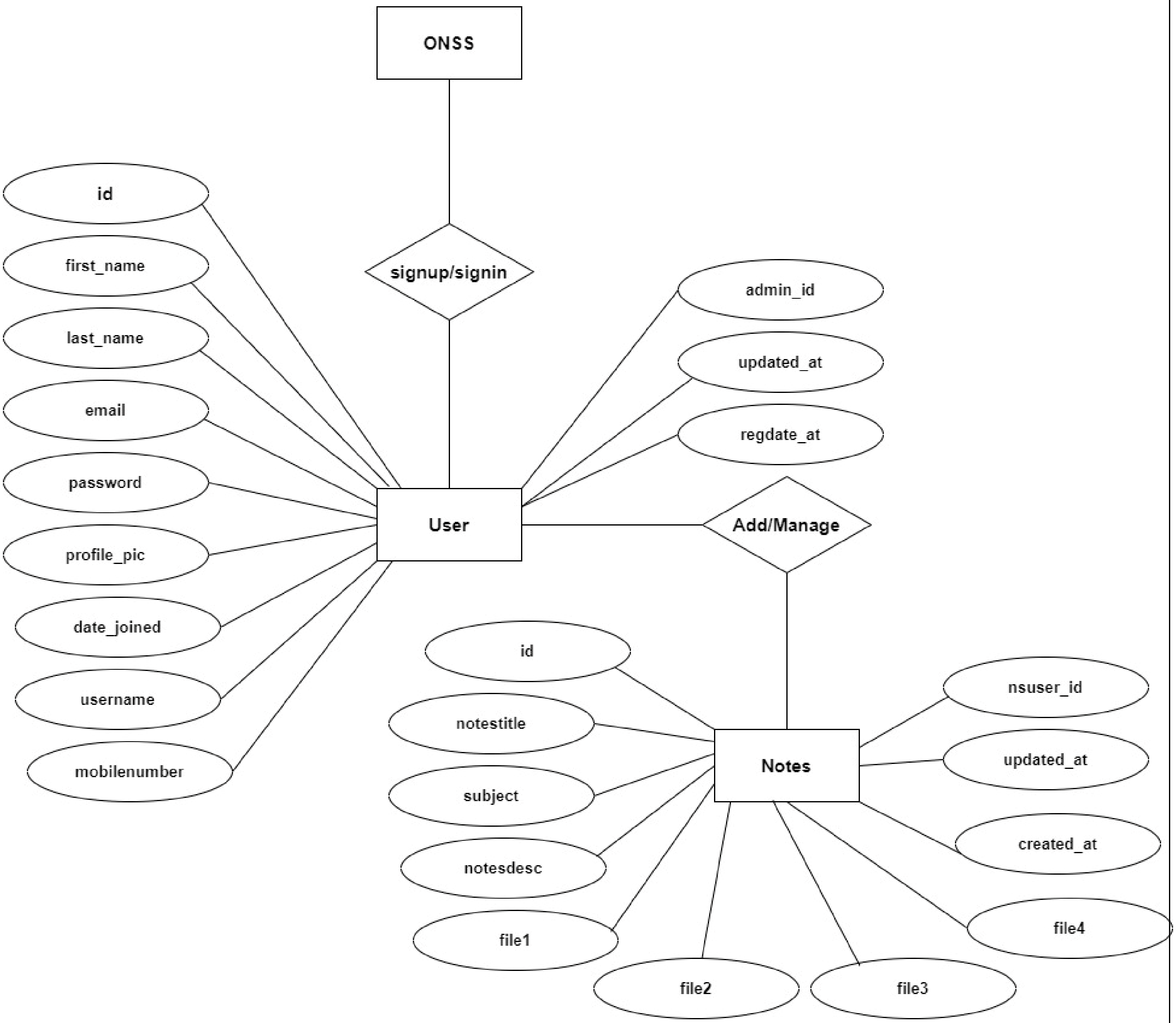
The symbols used in E-R diagrams are:

SYMBOL	PURPOSE
▭	Represents Entity sets.
⬭	Represent attributes.
◇	Represent Relationship Sets.
──	Line represents flow

Structured analysis is a set of tools and techniques that the analyst.

To develop a new kind of a system:

The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection a personal considerations.


ONSS
signup/signin
id
first_name
last_name
email
password
profile_pic
date_joined
username
mobilenumber
User
admin_id
updated_at
regdate_at
Add/Manage
Notes
id
notestitle
subject
notesdesc
file1
file2
file3
file4
nsuser_id
updated_at
created_at

Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

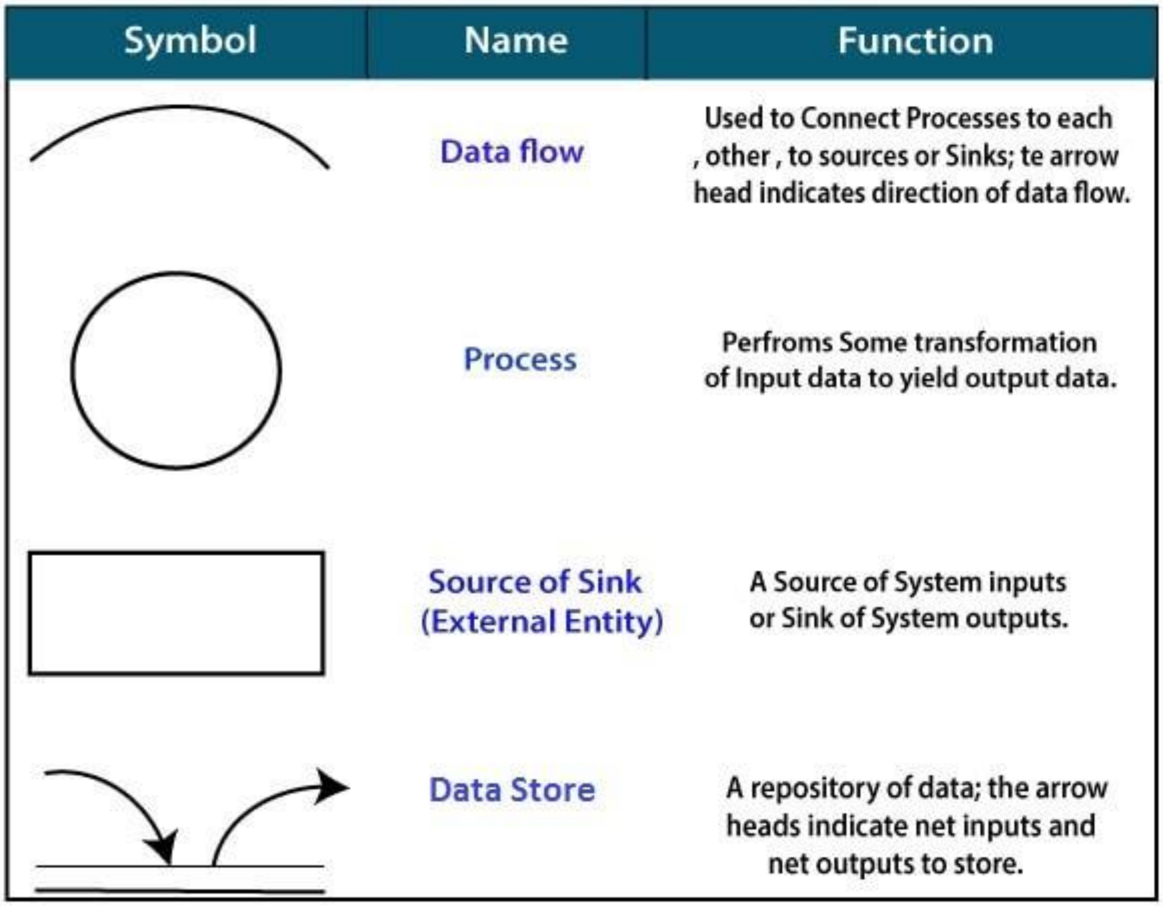
It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

The following observations about DFDs are essential:

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown in fig:



Symbol	Name	Function
	Data flow	Used to Connect Processes to each , other , to sources or Sinks; te arrow head indicates direction of data flow.
	Process	Perfroms Some transformation of Input data to yield output data.
	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

Symbols for Data Flow Diagrams

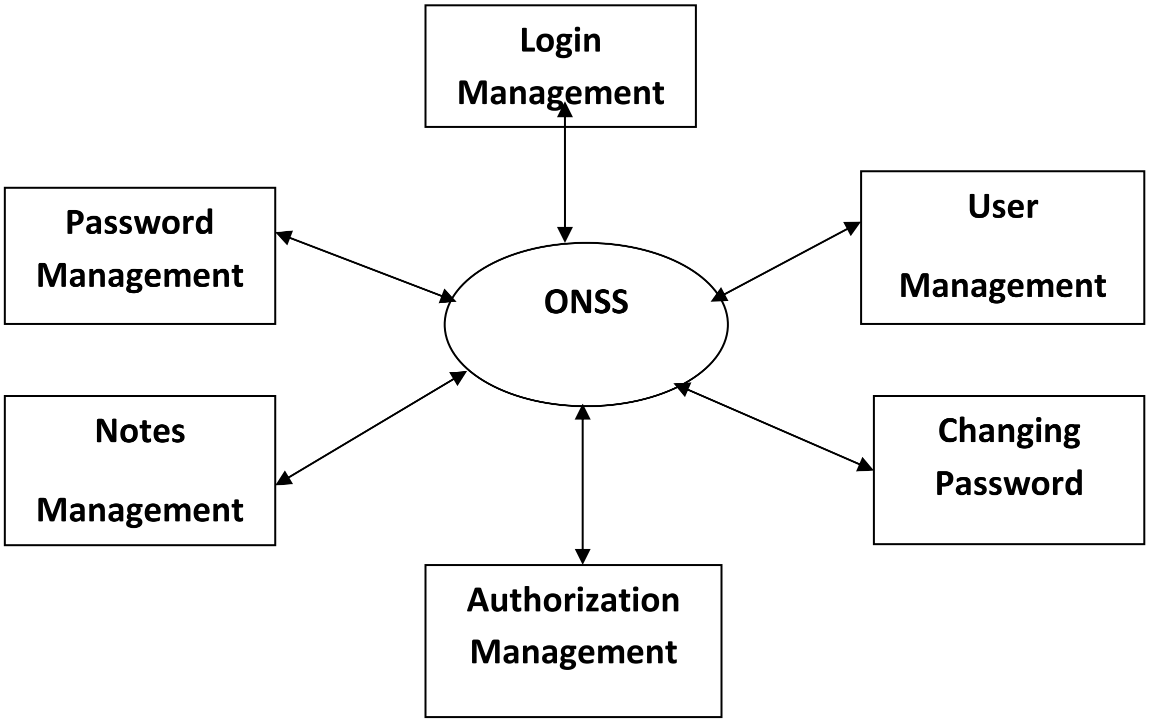
Circle: A circle (bubble) shows a process that transforms data inputs into data outputs.

Data Flow: A curved line shows the flow of data into or out of a process or data store.

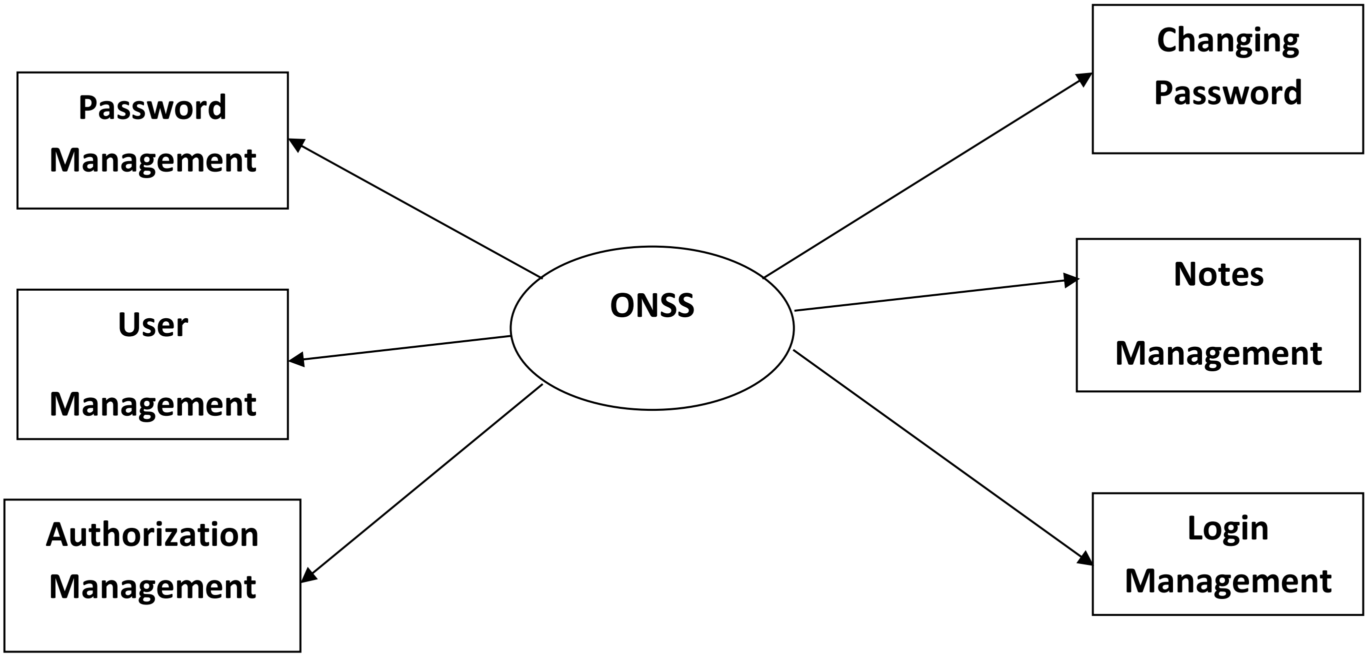
Data Store: A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

Source or Sink: Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

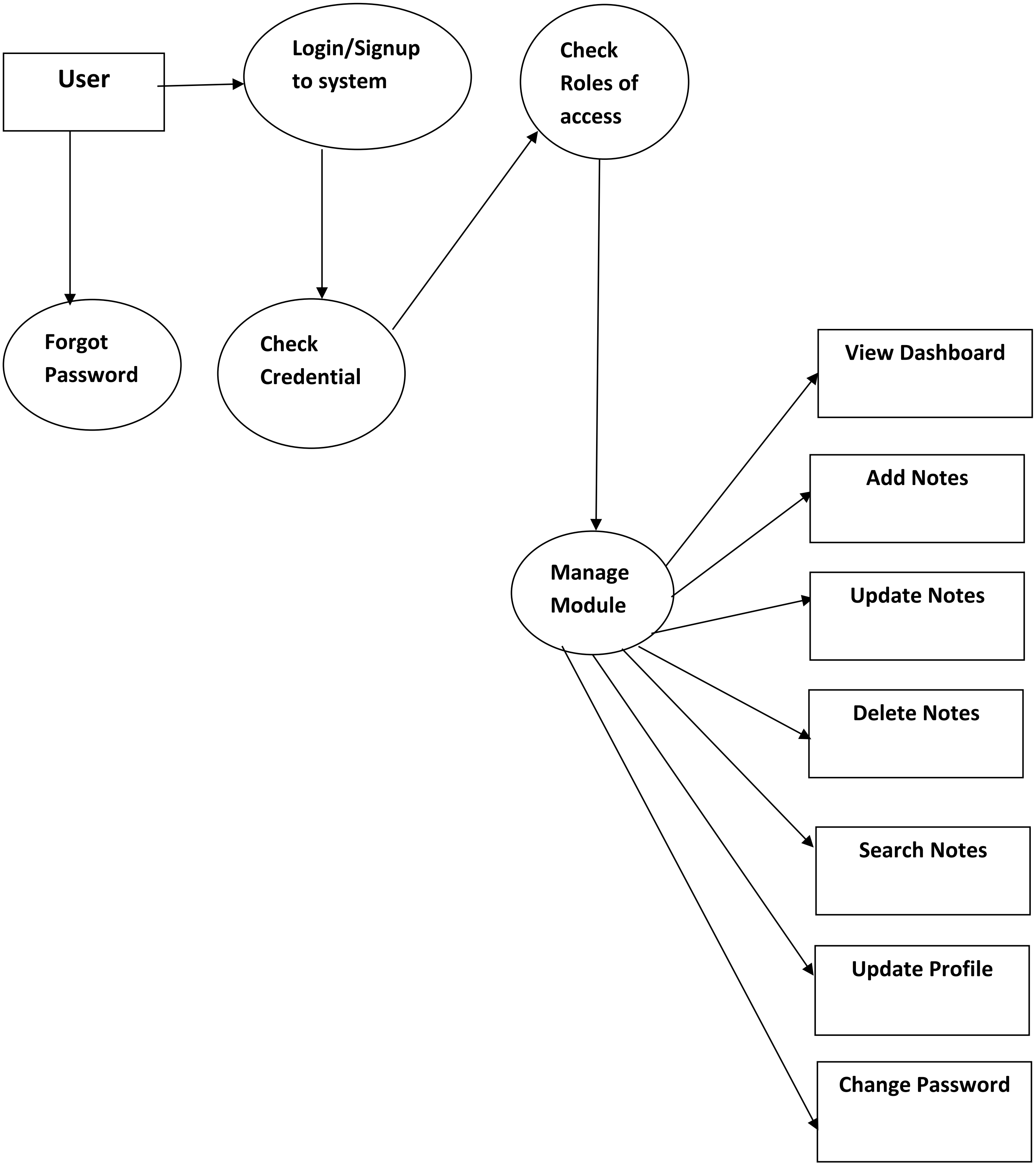
Zero Level DFD



First Level DFD



Second Level DFD



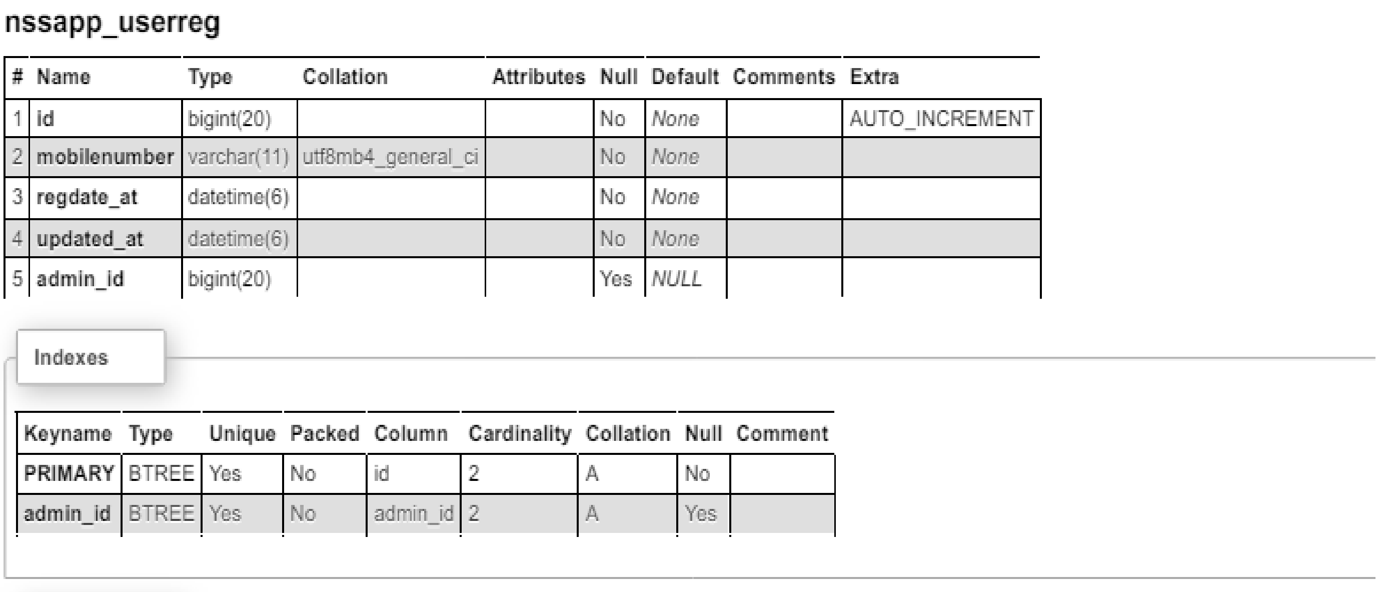
DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MySQL database has been chosen for developing the relevant databases.

Online Notes Sharing System contains 11 MySQL tables(In this MySQL 2 table is customized and 9 table made by default in django) :

User table Structure : This table store the login and personal Details of users.



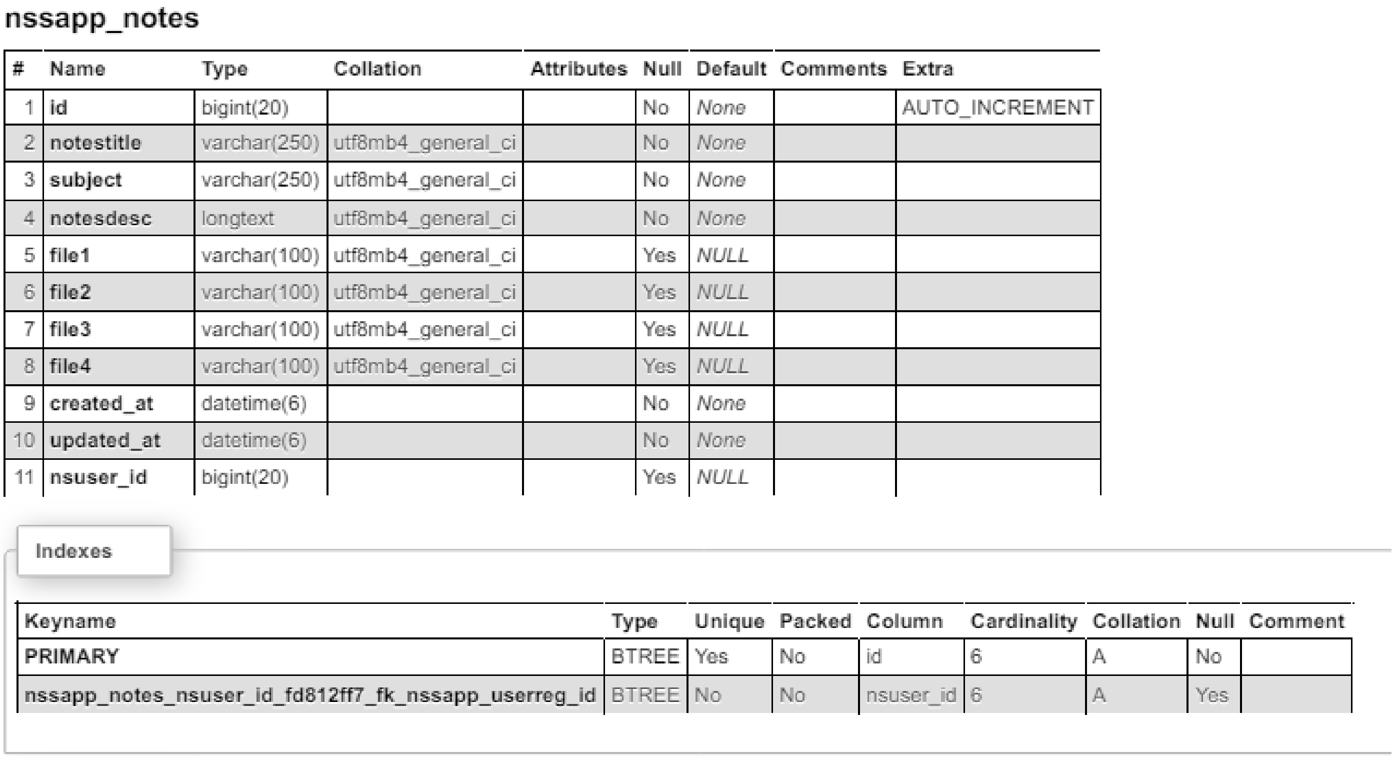
nssapp_userreg

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)			No	*None*		AUTO_INCREMENT
2	**mobilenumber**	varchar(11)	utf8mb4_general_ci		No	*None*		
3	**regdate_at**	datetime(6)			No	*None*		
4	**updated_at**	datetime(6)			No	*None*		
5	**admin_id**	bigint(20)			Yes	*NULL*		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	2	A	No	
admin_id	BTREE	Yes	No	admin_id	2	A	Yes	

Notes table Structure : This table store the details of notes which is share by users.



nssapp_notes

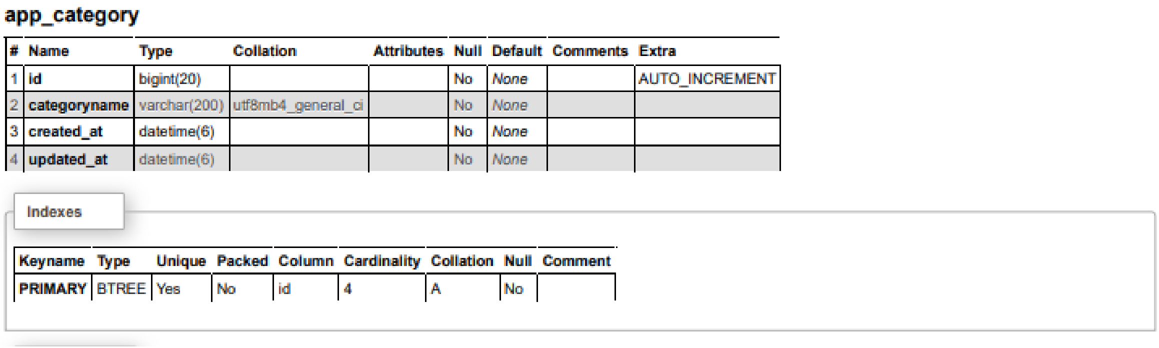
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)			No	*None*		AUTO_INCREMENT
2	**notestitle**	varchar(250)	utf8mb4_general_ci		No	*None*		
3	**subject**	varchar(250)	utf8mb4_general_ci		No	*None*		
4	**notesdesc**	longtext	utf8mb4_general_ci		No	*None*		
5	**file1**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
6	**file2**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
7	**file3**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
8	**file4**	varchar(100)	utf8mb4_general_ci		Yes	*NULL*		
9	**created_at**	datetime(6)			No	*None*		
10	**updated_at**	datetime(6)			No	*None*		
11	**nsuser_id**	bigint(20)			Yes	*NULL*		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	6	A	No	
nssapp_notes_nsuser_id_fd812ff7_fk_nssapp_userreg_id	BTREE	No	No	nsuser_id	6	A	Yes	

Customized Tables Details

app_category table Structure : This table store the category of bus.



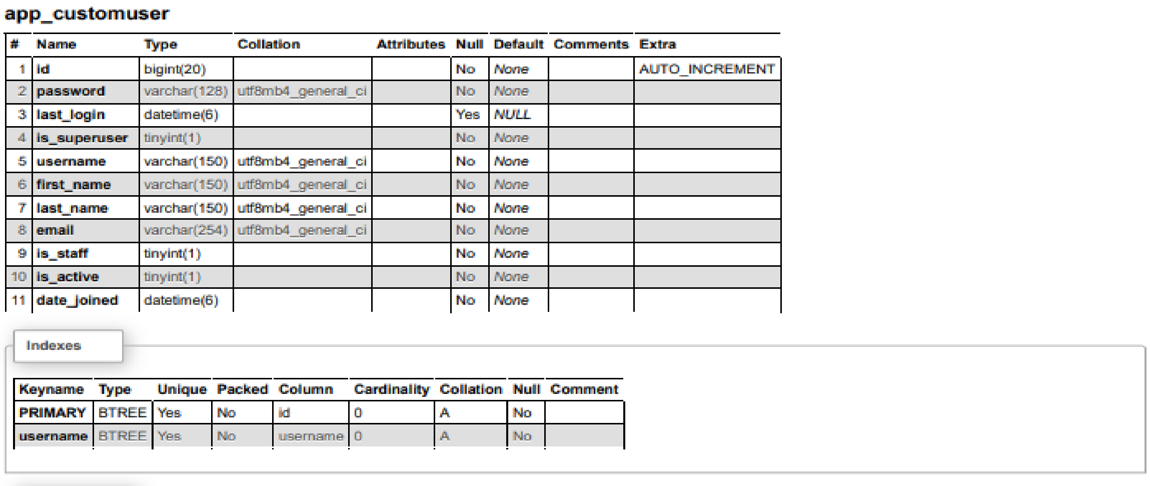
app_category

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	categoryname	varchar(200)	utf8mb4_general_ci		No	*None*		
3	created_at	datetime(6)			No	*None*		
4	updated_at	datetime(6)			No	*None*		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	4	A	No	

app_customuser table Structure : This table store the login and personal details of admin



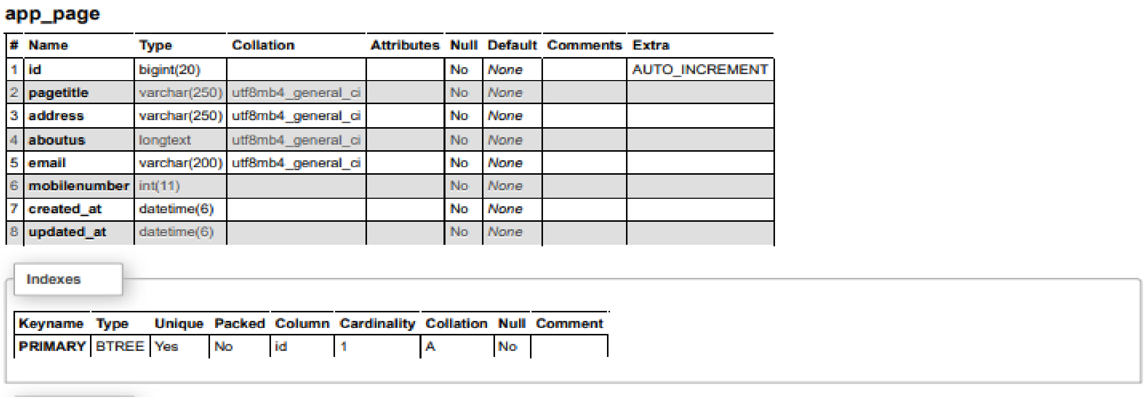
app_customuser

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	password	varchar(128)	utf8mb4_general_ci		No	*None*		
3	last_login	datetime(6)			Yes	*NULL*		
4	is_superuser	tinyint(1)			No	*None*		
5	username	varchar(150)	utf8mb4_general_ci		No	*None*		
6	first_name	varchar(150)	utf8mb4_general_ci		No	*None*		
7	last_name	varchar(150)	utf8mb4_general_ci		No	*None*		
8	email	varchar(254)	utf8mb4_general_ci		No	*None*		
9	is_staff	tinyint(1)			No	*None*		
10	is_active	tinyint(1)			No	*None*		
11	date_joined	datetime(6)			No	*None*		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	0	A	No	
username	BTREE	Yes	No	username	0	A	No	

app_page table Structure : This table store the details of website pages.



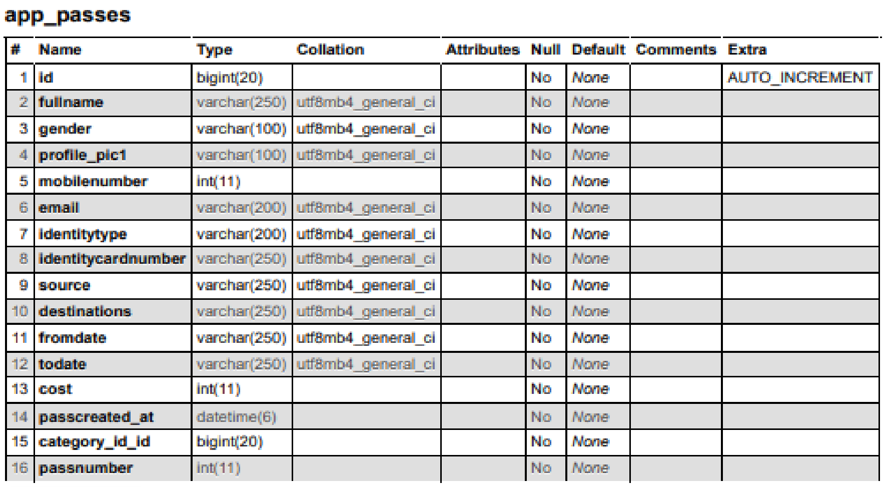
app_page

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	pagetitle	varchar(250)	utf8mb4_general_ci		No	*None*		
3	address	varchar(250)	utf8mb4_general_ci		No	*None*		
4	aboutus	longtext	utf8mb4_general_ci		No	*None*		
5	email	varchar(200)	utf8mb4_general_ci		No	*None*		
6	mobilenumber	int(11)			No	*None*		
7	created_at	datetime(6)			No	*None*		
8	updated_at	datetime(6)			No	*None*		

Indexes

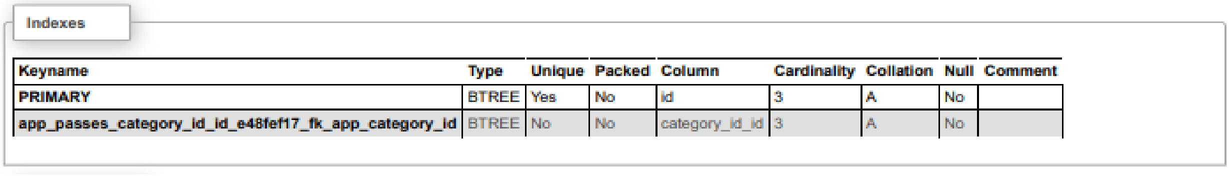
Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	1	A	No	

app_passes table Structure : This table store the pass details of person.



app_passes

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	fullname	varchar(250)	utf8mb4_general_ci		No	*None*		
3	gender	varchar(100)	utf8mb4_general_ci		No	*None*		
4	profile_pic1	varchar(100)	utf8mb4_general_ci		No	*None*		
5	mobilenumber	int(11)			No	*None*		
6	email	varchar(200)	utf8mb4_general_ci		No	*None*		
7	identitytype	varchar(200)	utf8mb4_general_ci		No	*None*		
8	identitycardnumber	varchar(250)	utf8mb4_general_ci		No	*None*		
9	source	varchar(250)	utf8mb4_general_ci		No	*None*		
10	destinations	varchar(250)	utf8mb4_general_ci		No	*None*		
11	fromdate	varchar(250)	utf8mb4_general_ci		No	*None*		
12	todate	varchar(250)	utf8mb4_general_ci		No	*None*		
13	cost	int(11)			No	*None*		
14	passcreated_at	datetime(6)			No	*None*		
15	category_id_id	bigint(20)			No	*None*		
16	passnumber	int(11)			No	*None*		

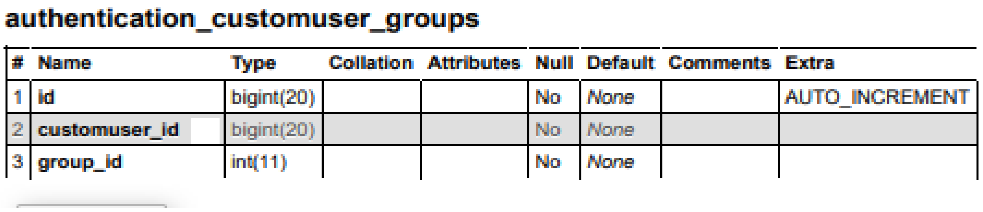


Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	3	A	No	
app_passes_category_id_id_e48fef17_fk_app_category_id	BTREE	No	No	category_id_id	3	A	No	

Default Tables Details

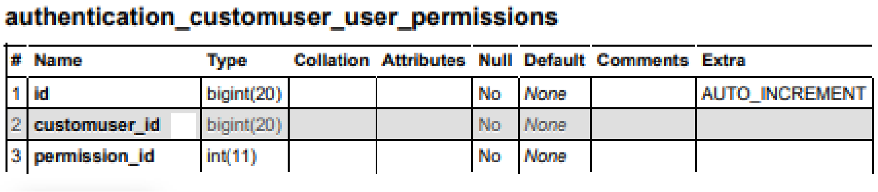
authentication_customuser_groups table Structure : This table by default table.



authentication_customuser_groups

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	customuser_id	bigint(20)			No	*None*		
3	group_id	int(11)			No	*None*		

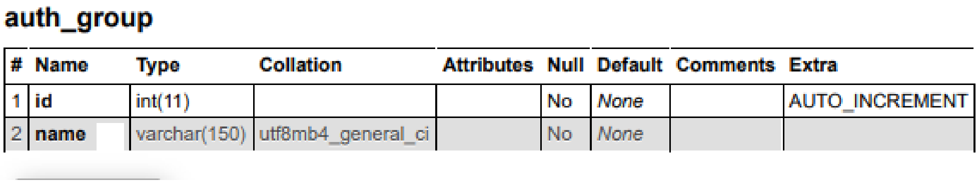
authentication_customuser_user_permissions table Structure : This table by default table.



authentication_customuser_user_permissions

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	customuser_id	bigint(20)			No	*None*		
3	permission_id	int(11)			No	*None*		

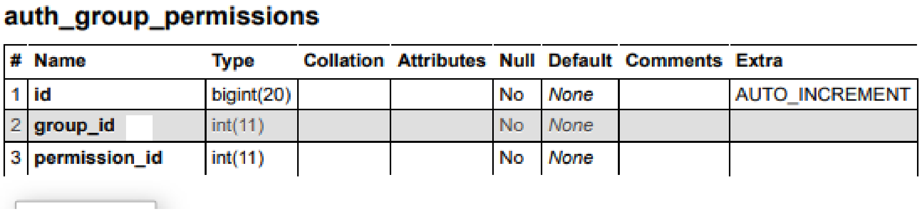
auth_group table Structure : This table in Django is part of the Django authentication system and is used to represent groups of users.



auth_group

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	name	varchar(150)	utf8mb4_general_ci		No	*None*		

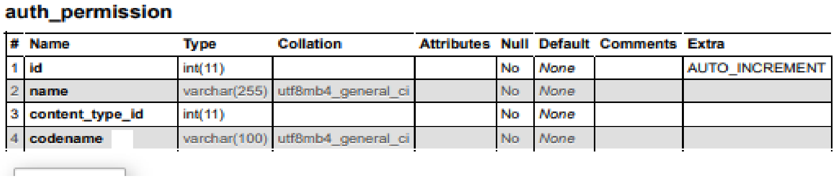
auth_group_permissions table Structure : This table in Django is a part of the permission system and is used to manage the relationship between user groups and permissions.



auth_group_permissions

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	bigint(20)			No	*None*		AUTO_INCREMENT
2	group_id	int(11)			No	*None*		
3	permission_id	int(11)			No	*None*		

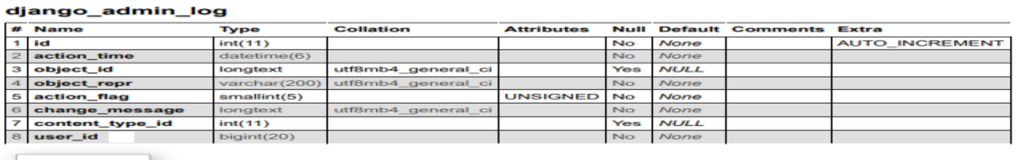
auth_permission table Structure : This table in Django is used to store information about the permissions defined.



auth_permission

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_general_ci		No	*None*		
3	content_type_id	int(11)			No	*None*		
4	codename	varchar(100)	utf8mb4_general_ci		No	*None*		

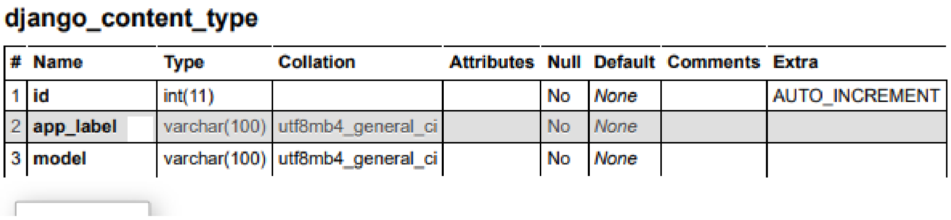
django_admin_log table Structure : This table is used to store records of actions taken by administrators or users through the Django admin interface.



django_admin_log

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	*None*		AUTO_INCREMENT
2	action_time	datetime(6)			No	*None*		
3	object_id	longtext	utf8mb4_general_ci		Yes	*NULL*		
4	object_repr	varchar(200)	utf8mb4_general_ci		No	*None*		
5	action_flag	smallint(5)		UNSIGNED	No	*None*		
6	change_message	longtext	utf8mb4_general_ci		No	*None*		
7	content_type_id	int(11)			Yes	*NULL*		
8	user_id	bigint(20)			No	*None*		

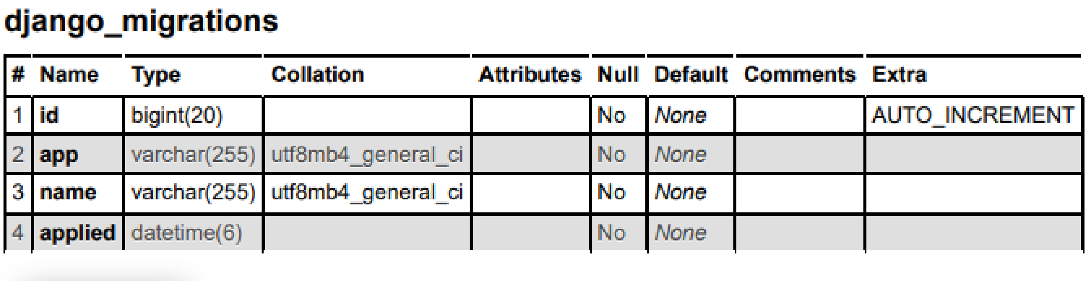
django_content_type table Structure : This table in Django is a system table that is used to store information about each model (database table) in your Django project.



django_content_type

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int(11)			No	*None*		AUTO_INCREMENT
2	**app_label**	varchar(100)	utf8mb4_general_ci		No	*None*		
3	**model**	varchar(100)	utf8mb4_general_ci		No	*None*		

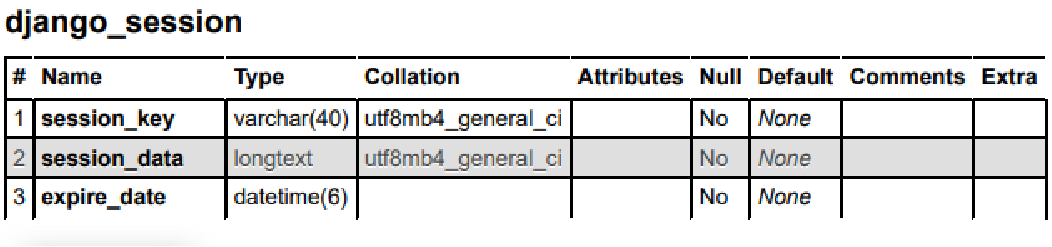
django_migrations table Structure : This table table is part of the database schema and is used to keep track of which migrations have been applied to the database.



django_migrations

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	bigint(20)			No	*None*		AUTO_INCREMENT
2	**app**	varchar(255)	utf8mb4_general_ci		No	*None*		
3	**name**	varchar(255)	utf8mb4_general_ci		No	*None*		
4	**applied**	datetime(6)			No	*None*		

django_session table Structure : This table store the session data for user.

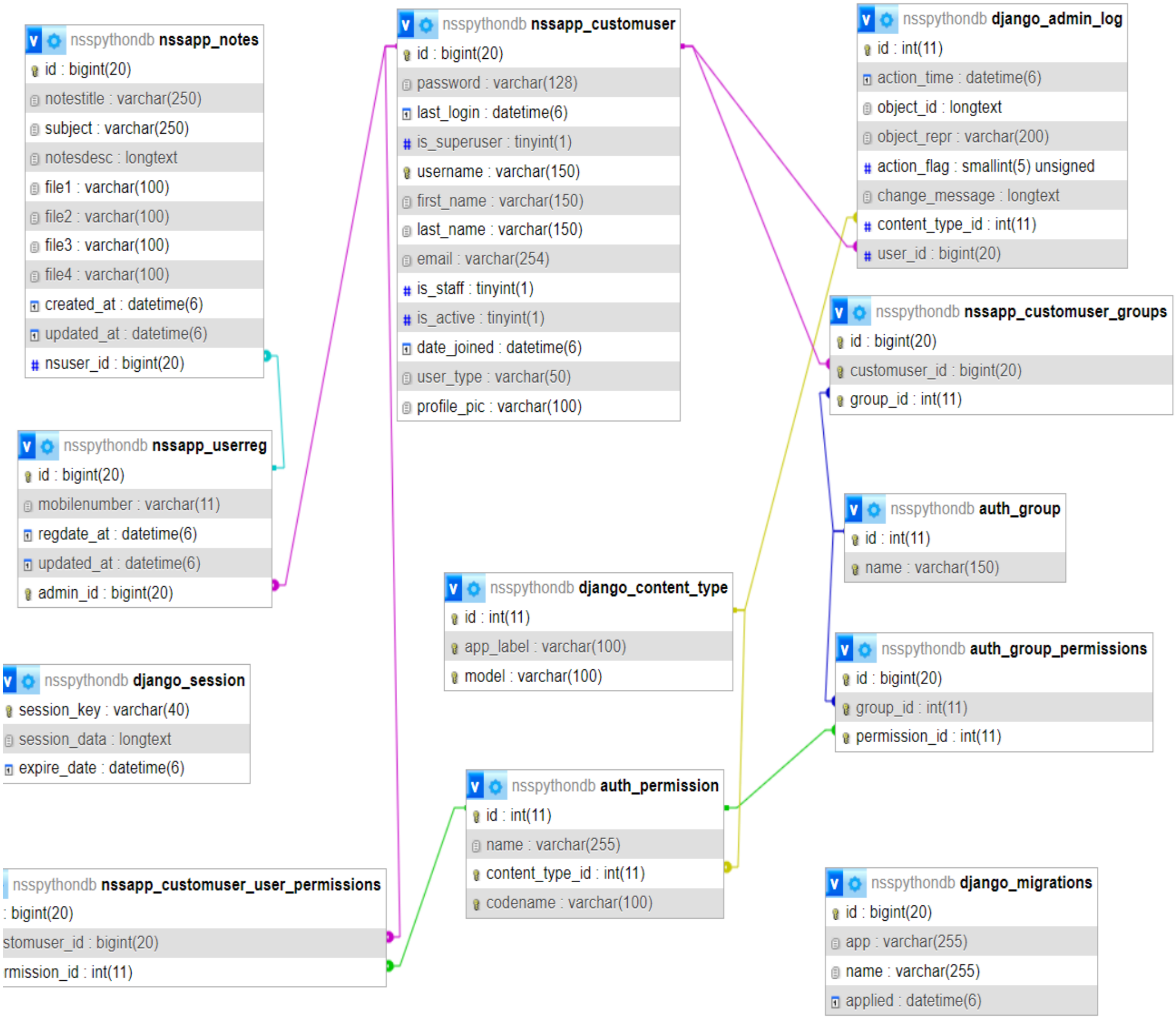


django_session

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**session_key**	varchar(40)	utf8mb4_general_ci		No	*None*		
2	**session_data**	longtext	utf8mb4_general_ci		No	*None*		
3	**expire_date**	datetime(6)			No	*None*		

Class Diagram:

The class diagram shows a set of classes, interfaces, collaborations and their relationships.



nsspythondb **nssapp_notes**
- id : bigint(20)
- notestitle : varchar(250)
- subject : varchar(250)
- notesdesc : longtext
- file1 : varchar(100)
- file2 : varchar(100)
- file3 : varchar(100)
- file4 : varchar(100)
- created_at : datetime(6)
- updated_at : datetime(6)
- nsuser_id : bigint(20)

nsspythondb **nssapp_userreg**
- id : bigint(20)
- mobilenumber : varchar(11)
- regdate_at : datetime(6)
- updated_at : datetime(6)
- admin_id : bigint(20)

nsspythondb **django_session**
- session_key : varchar(40)
- session_data : longtext
- expire_date : datetime(6)

nsspythondb **nssapp_customuser_user_permissions**
- : bigint(20)
- stomuser_id : bigint(20)
- rmission_id : int(11)

nsspythondb **nssapp_customuser**
- id : bigint(20)
- password : varchar(128)
- last_login : datetime(6)
- is_superuser : tinyint(1)
- username : varchar(150)
- first_name : varchar(150)
- last_name : varchar(150)
- email : varchar(254)
- is_staff : tinyint(1)
- is_active : tinyint(1)
- date_joined : datetime(6)
- user_type : varchar(50)
- profile_pic : varchar(100)

nsspythondb **django_content_type**
- id : int(11)
- app_label : varchar(100)
- model : varchar(100)

nsspythondb **auth_permission**
- id : int(11)
- name : varchar(255)
- content_type_id : int(11)
- codename : varchar(100)

nsspythondb **django_admin_log**
- id : int(11)
- action_time : datetime(6)
- object_id : longtext
- object_repr : varchar(200)
- action_flag : smallint(5) unsigned
- change_message : longtext
- content_type_id : int(11)
- user_id : bigint(20)

nsspythondb **nssapp_customuser_groups**
- id : bigint(20)
- customuser_id : bigint(20)
- group_id : int(11)

nsspythondb **auth_group**
- id : int(11)
- name : varchar(150)

nsspythondb **auth_group_permissions**
- id : bigint(20)
- group_id : int(11)
- permission_id : int(11)

nsspythondb **django_migrations**
- id : bigint(20)
- app : varchar(255)
- name : varchar(255)
- applied : datetime(6)

SYSTEM TESTING

SOFTWARE TESTING TECHNIQUES:

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

TESTING OBJECTIVES:

1. Testing is process of executing a program with the intent of finding an error.

2. A good test case design is one that has a probability of finding an as yet undiscovered error.

3. A successful test is one that uncovers an as yet undiscovered error.

These above objectives imply a dramatic change in view port.

Testing cannot show the absence of defects, it can only show that software errors are present.

There are three types of testing strategies

1. Unit test
2. Integration test
3. Performance test

Unit Testing:

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

Integration Testing:

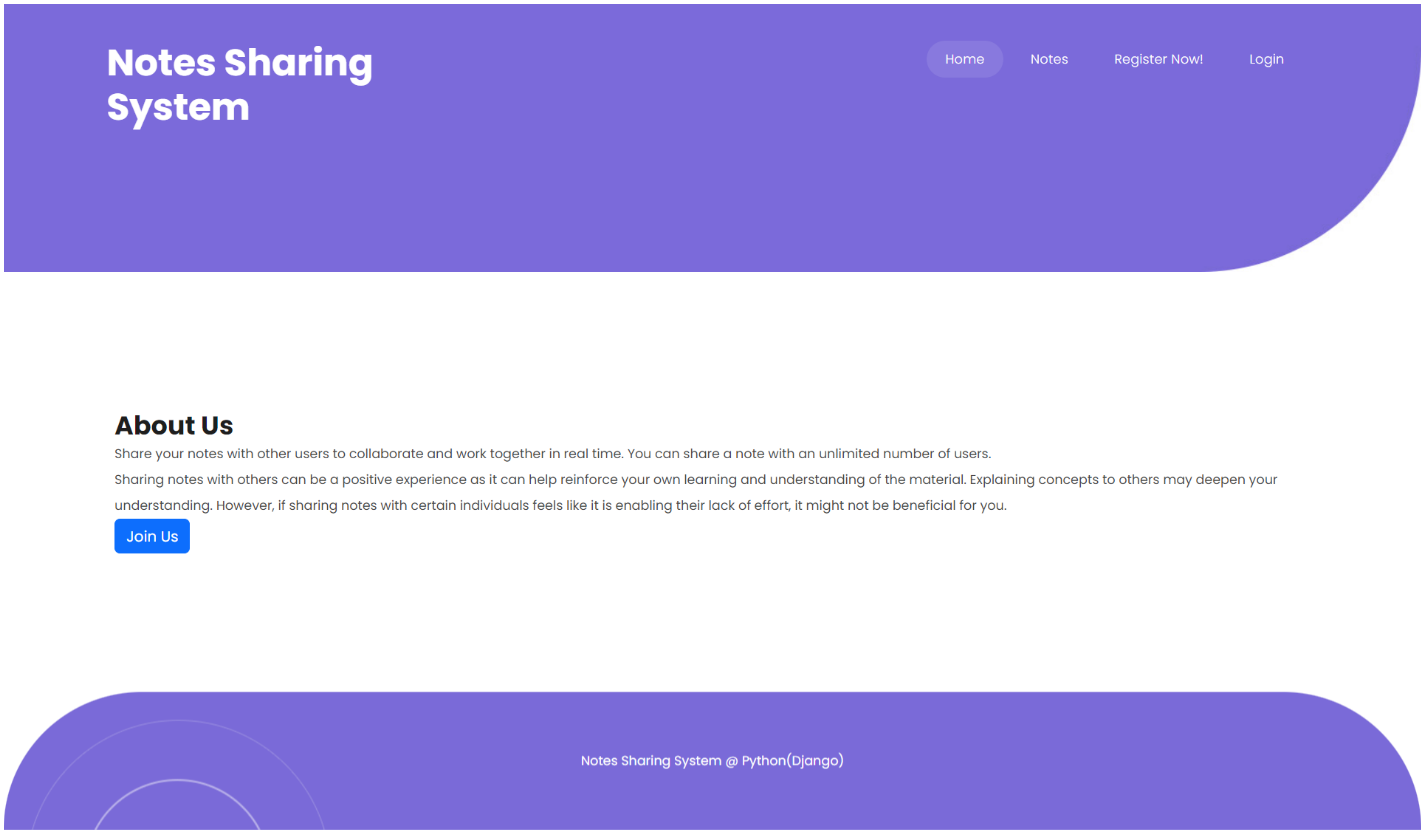
Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

Performance Testing:

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

OUTPUT SCREEN OF PROJECT

Home Page

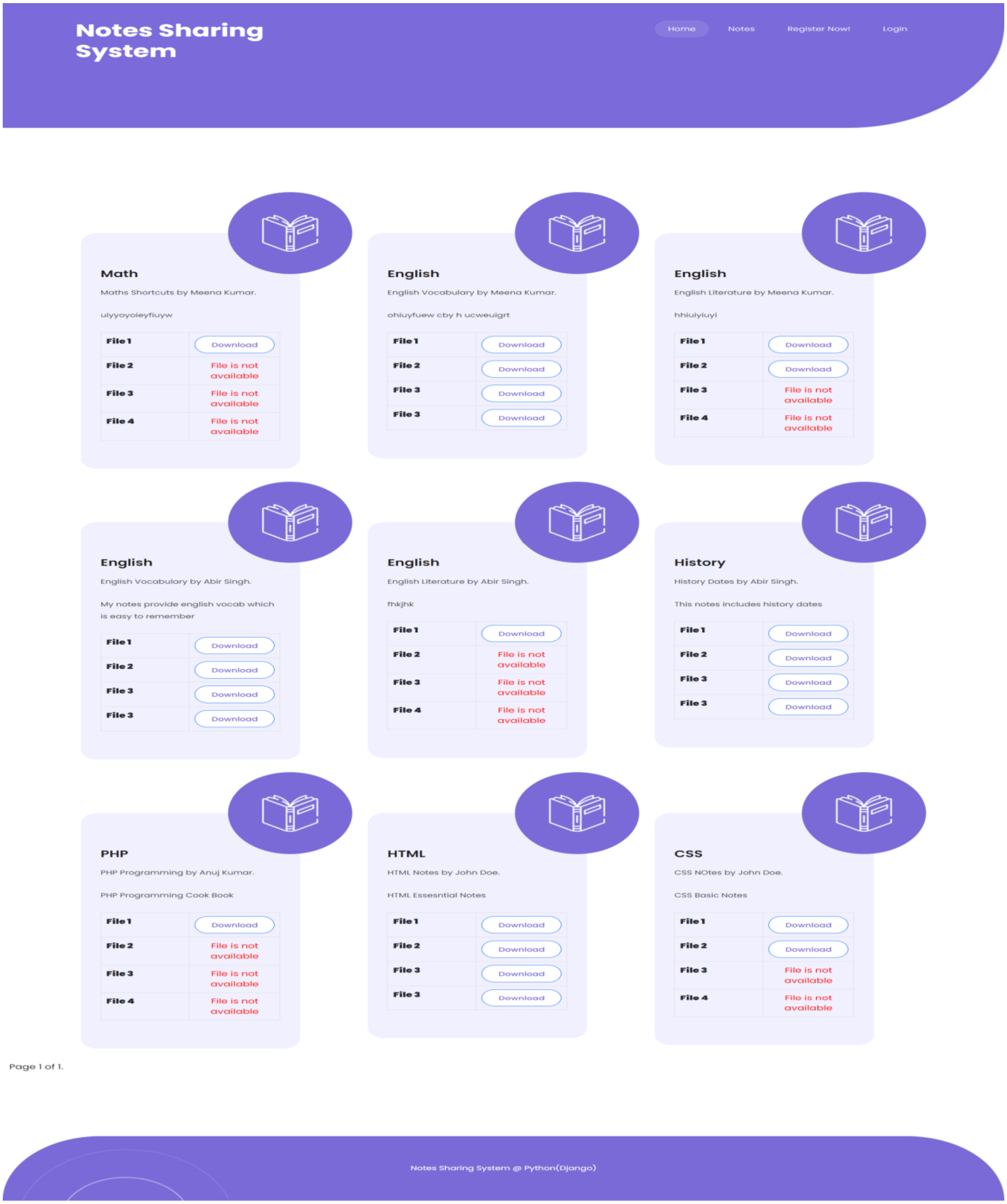


About Us

Share your notes with other users to collaborate and work together in real time. You can share a note with an unlimited number of users.

Sharing notes with others can be a positive experience as it can help reinforce your own learning and understanding of the material. Explaining concepts to others may deepen your understanding. However, if sharing notes with certain individuals feels like it is enabling their lack of effort, it might not be beneficial for you.

Notes Details



Math

Maths Shortcuts by Meena Kumar.

uiyyoyoieyfiuyw

File 1	Download
File 2	File is not available
File 3	File is not available
File 4	File is not available

English

English Vocabulary by Meena Kumar.

ohiuyfuew cby h ucweuigrt

File 1	Download
File 2	Download
File 3	Download
File 3	Download

English

English Literature by Meena Kumar.

hhiuiyiuyi

File 1	Download
File 2	Download
File 3	File is not available
File 4	File is not available

English

English Vocabulary by Abir Singh.

My notes provide english vocab which is easy to remember

File 1	Download
File 2	Download
File 3	Download
File 3	Download

English

English Literature by Abir Singh.

fhkjhk

File 1	Download
File 2	File is not available
File 3	File is not available
File 4	File is not available

History

History Dates by Abir Singh.

This notes includes history dates

File 1	Download
File 2	Download
File 3	Download
File 3	Download

PHP

PHP Programming by Anuj Kumar.

PHP Programming Cook Book

File 1	Download
File 2	File is not available
File 3	File is not available
File 4	File is not available

HTML

HTML Notes by John Doe.

HTML Essesntial Notes

File 1	Download
File 2	Download
File 3	Download
File 3	Download

CSS

CSS NOtes by John Doe.

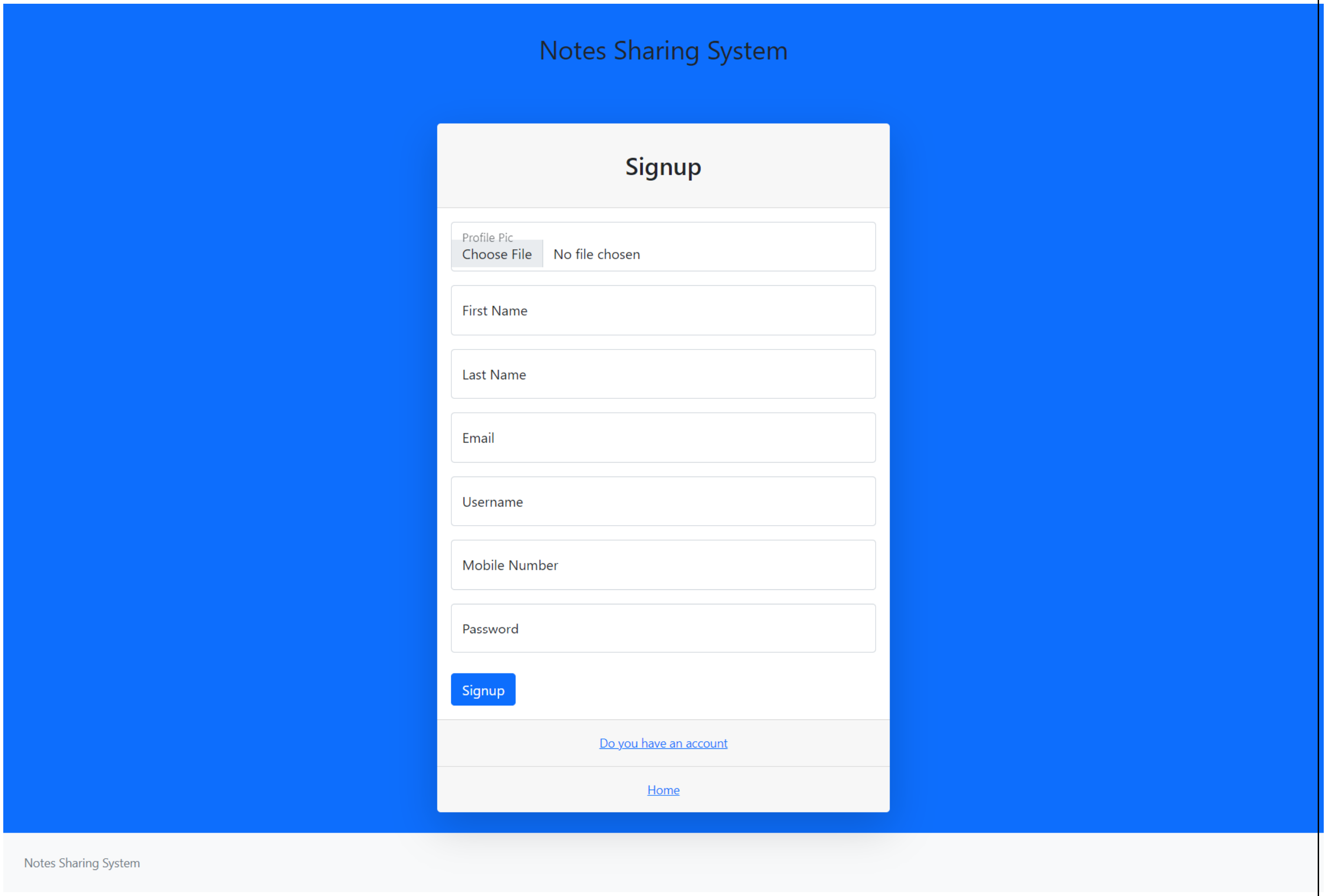
CSS Basic Notes

File 1	Download
File 2	Download
File 3	File is not available
File 4	File is not available

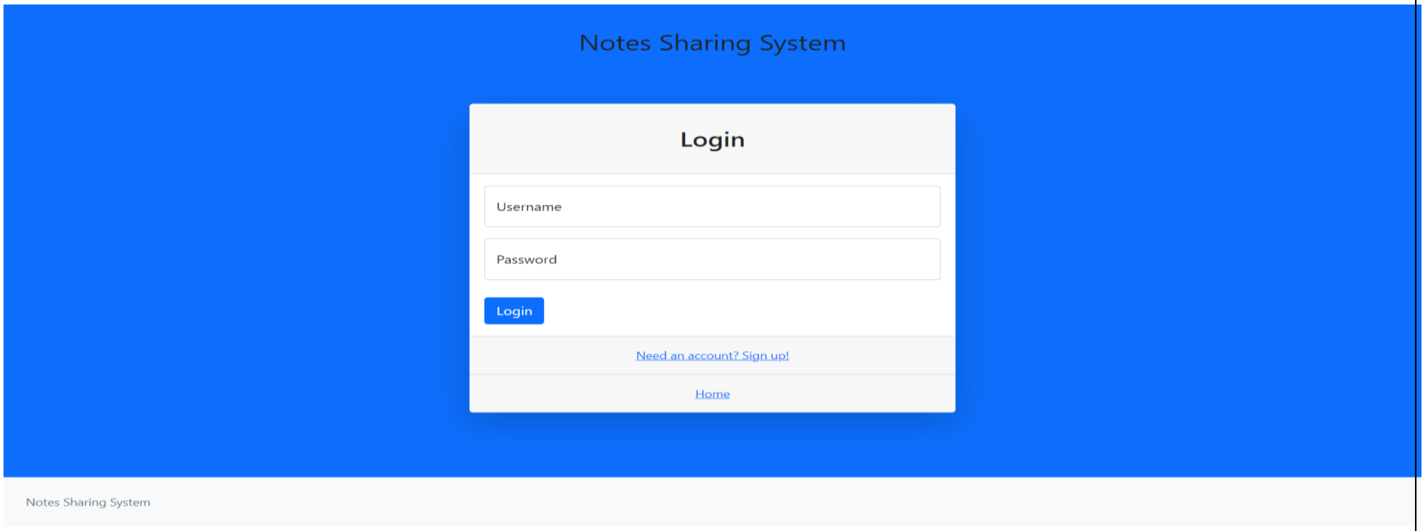
Page 1 of 1.

Notes Sharing System @ Python(Django)

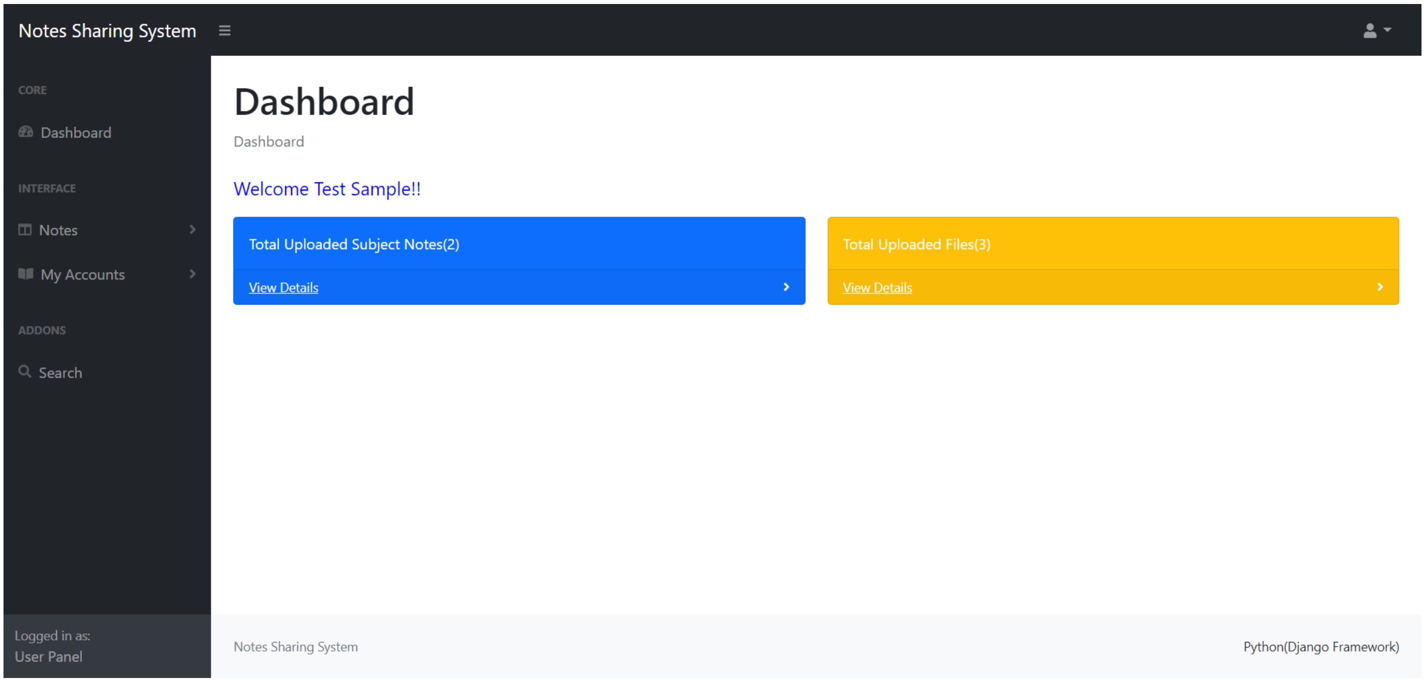
User Signup



User Sign in



Dashboard



Dashboard

Dashboard

Welcome Test Sample!!

Total Uploaded Subject Notes(2)

View Details

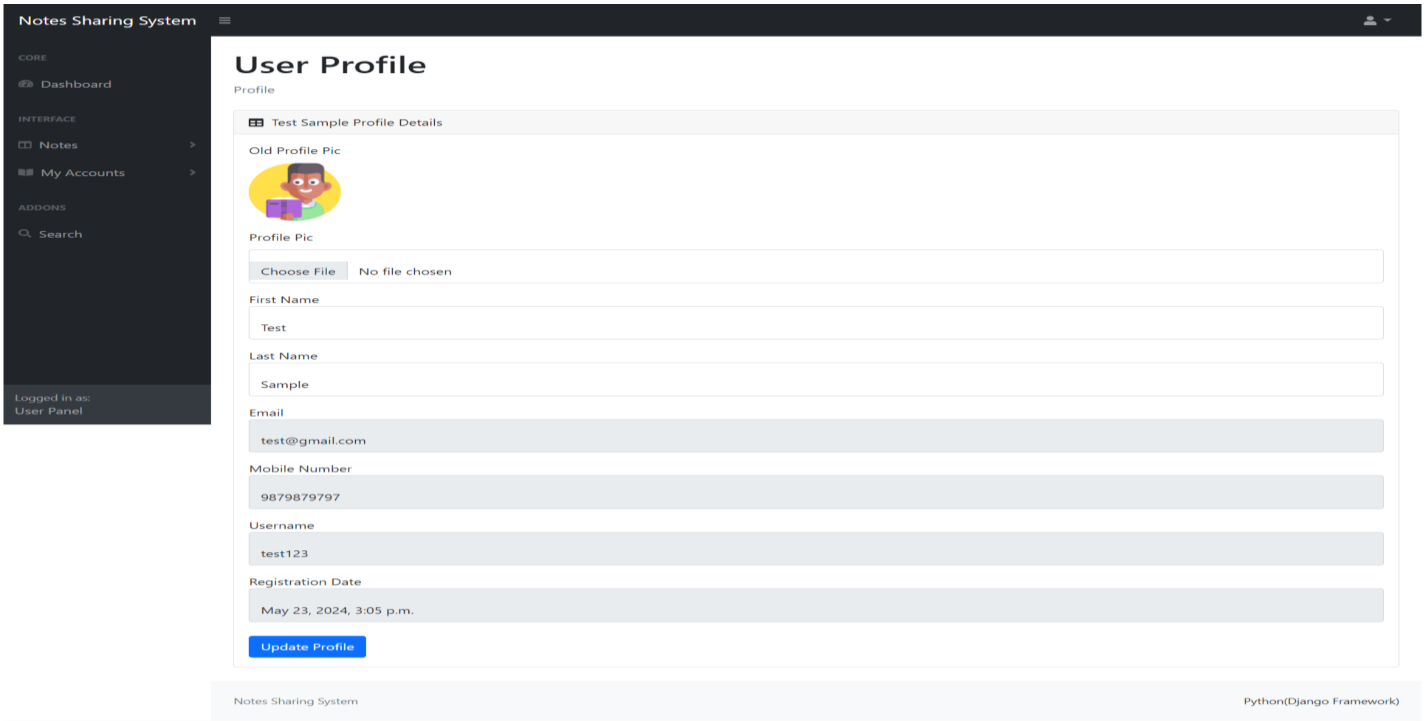
Total Uploaded Files(3)

View Details

Notes Sharing System

Python(Django Framework)

Profile



User Profile

Profile

Test Sample Profile Details

Old Profile Pic

Profile Pic: No file chosen

First Name: Test

Last Name: Sample

Email: test@gmail.com

Mobile Number: 9879879797

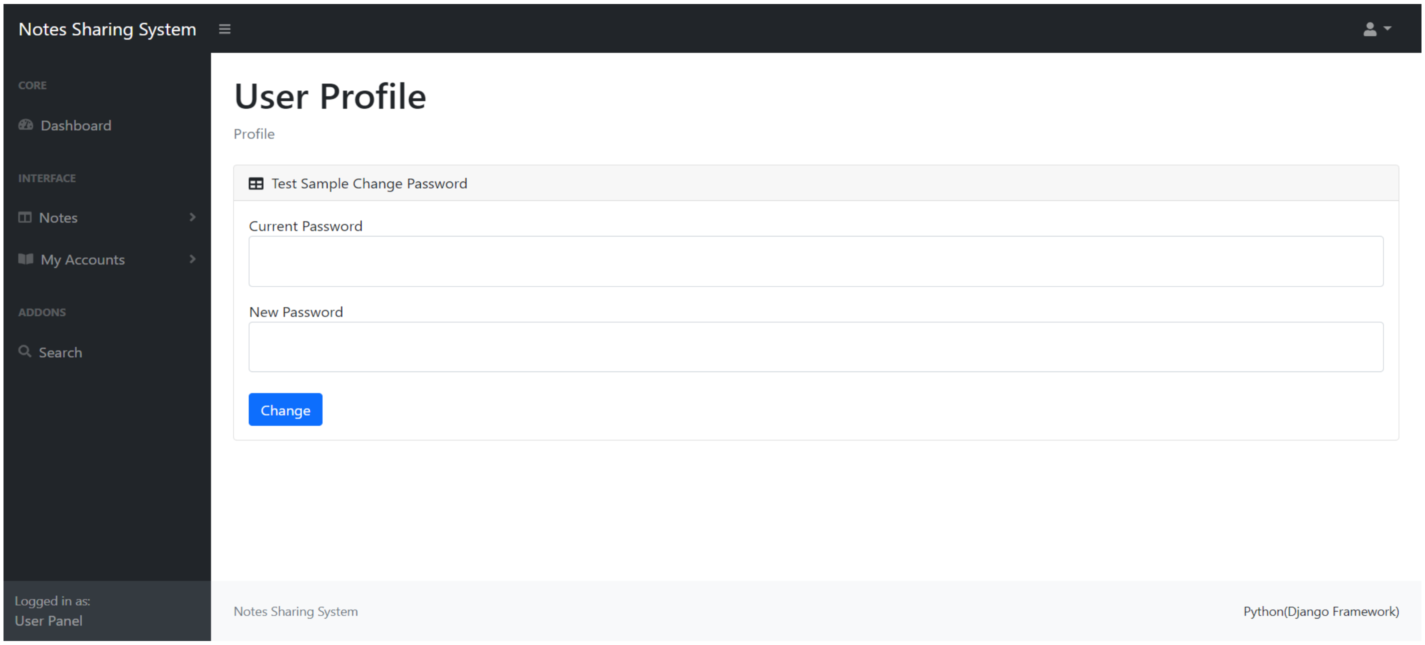
Username: test123

Registration Date: May 23, 2024, 3:05 p.m.

Notes Sharing System

Python(Django Framework)

Change Password



User Profile

Profile

Test Sample Change Password

Current Password

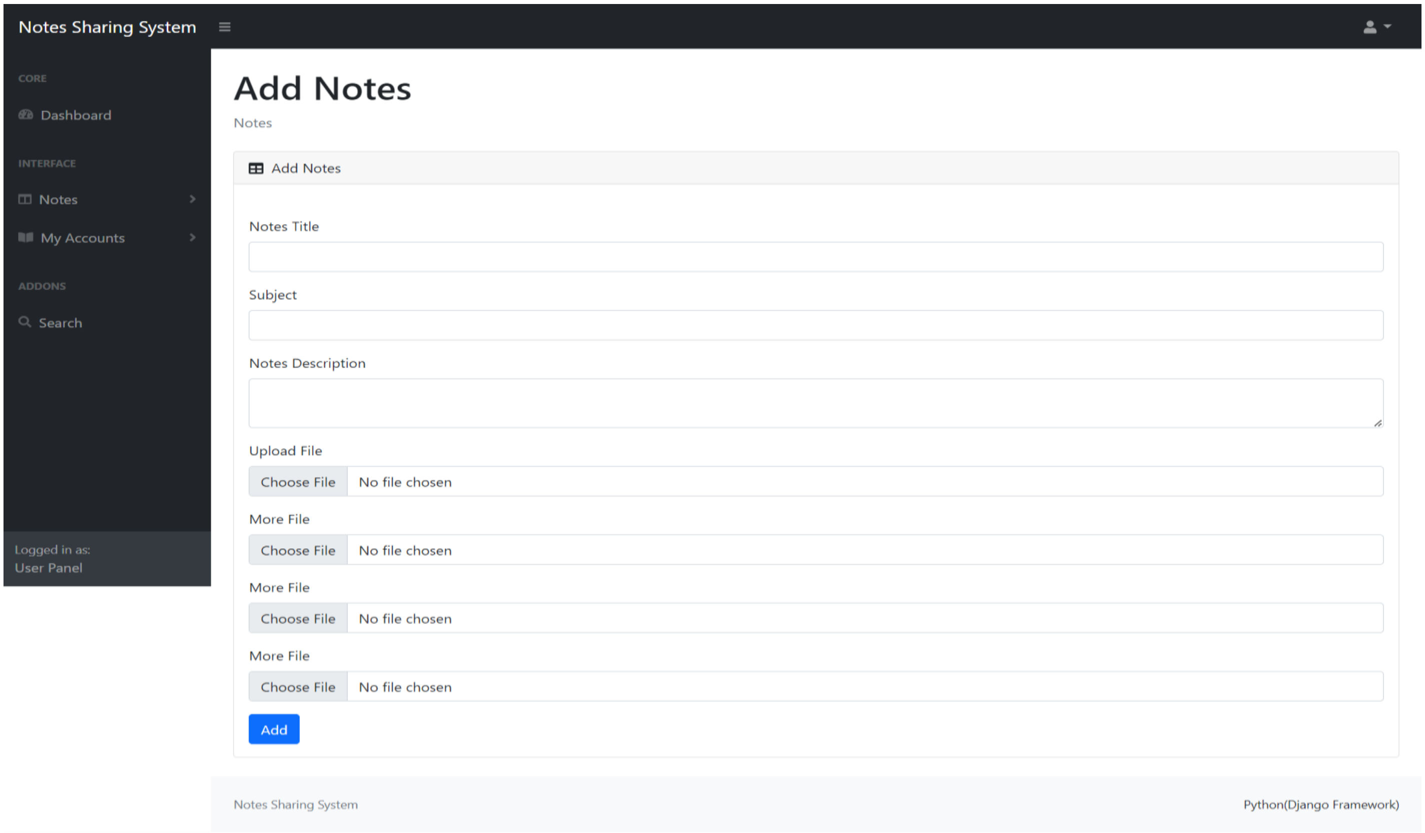
New Password

Change

Notes Sharing System

Python(Django Framework)

Add Notes



Add Notes

Notes

Add Notes

Notes Title

Subject

Notes Description

Upload File

Choose File No file chosen

More File

Choose File No file chosen

More File

Choose File No file chosen

More File

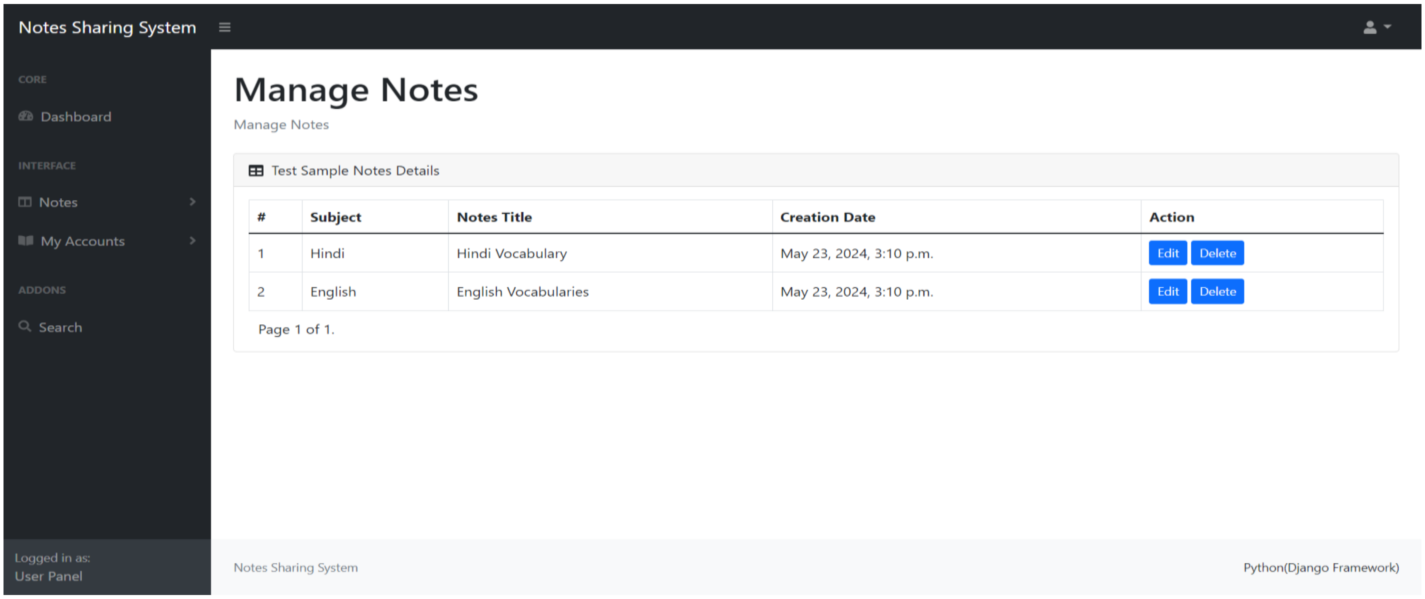
Choose File No file chosen

Add

Notes Sharing System

Python(Django Framework)

Manage Notes



Manage Notes

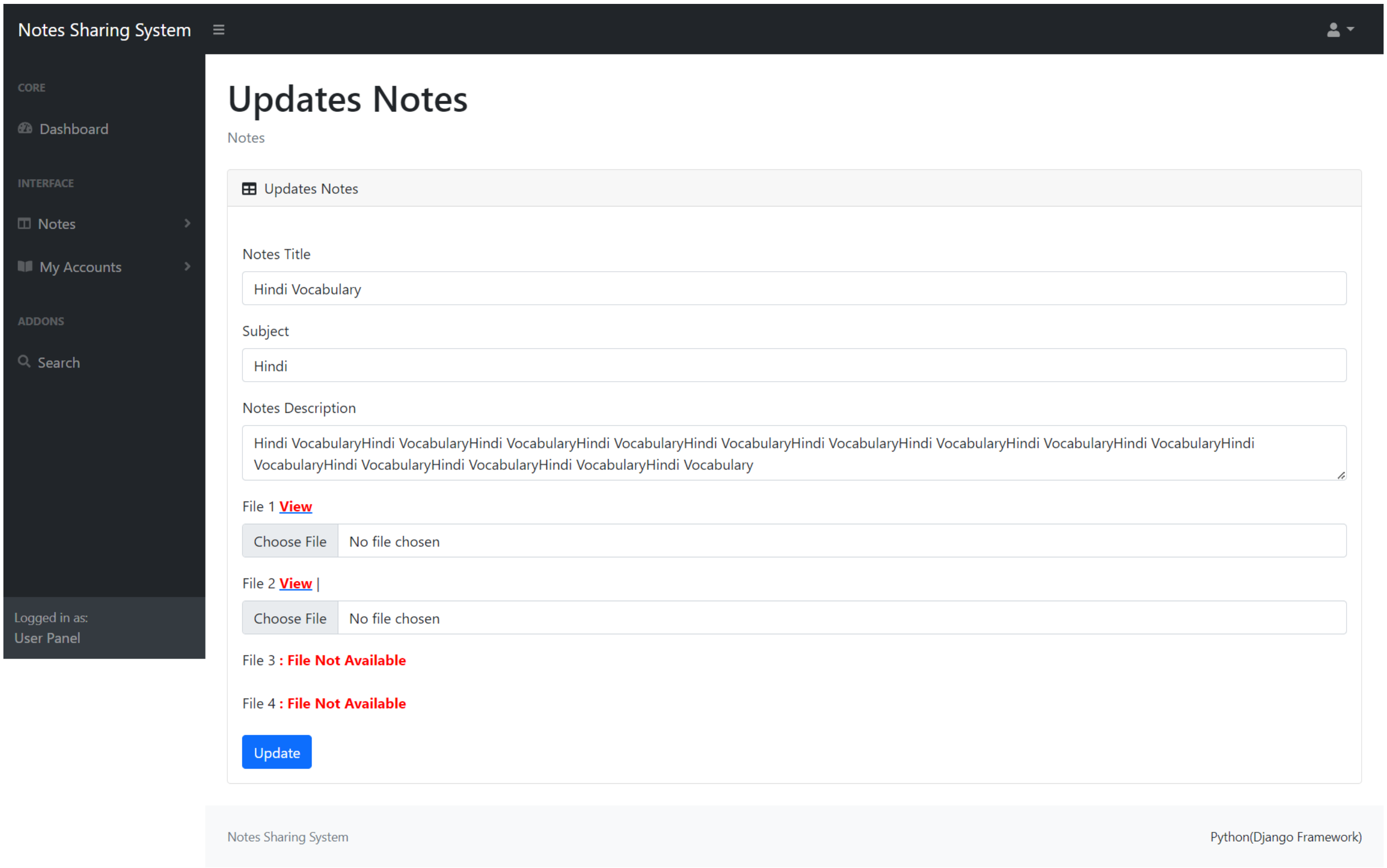
Manage Notes

Test Sample Notes Details

#	Subject	Notes Title	Creation Date	Action
1	Hindi	Hindi Vocabulary	May 23, 2024, 3:10 p.m.	Edit Delete
2	English	English Vocabularies	May 23, 2024, 3:10 p.m.	Edit Delete

Page 1 of 1.

Update Notes



Updates Notes

Notes

Updates Notes

Notes Title

Hindi Vocabulary

Subject

Hindi

Notes Description

Hindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi VocabularyHindi Vocabulary

File 1 **View**

Choose File No file chosen

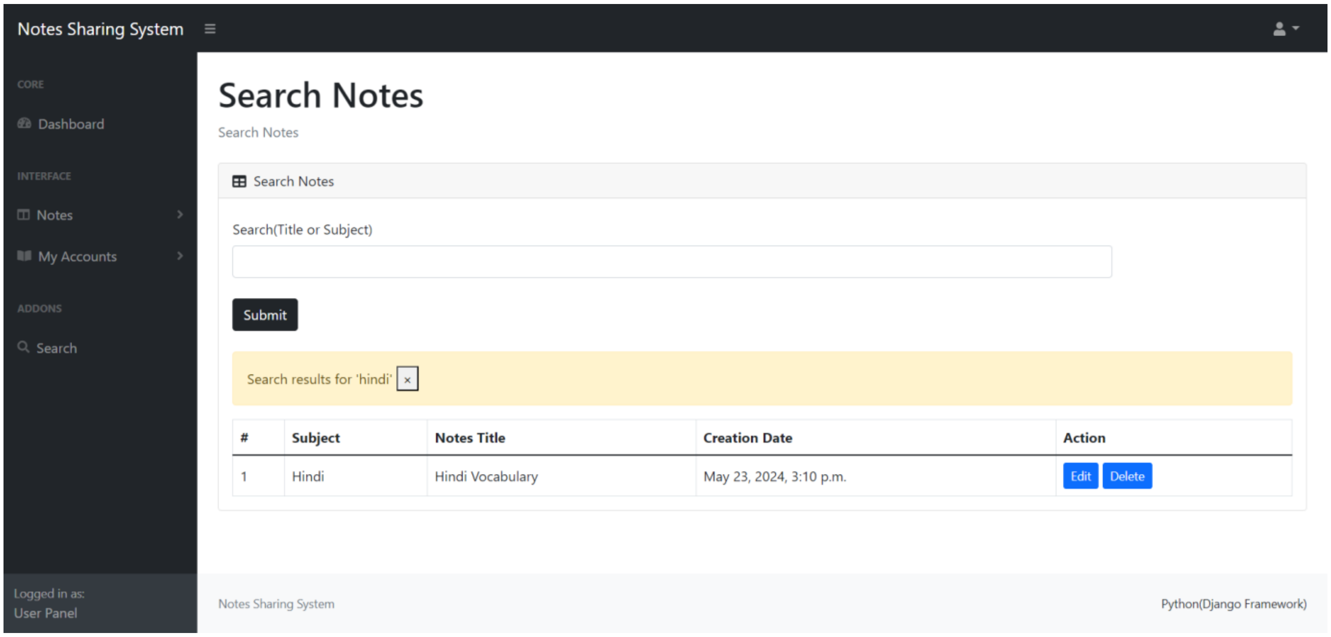
File 2 **View** |

Choose File No file chosen

File 3 : **File Not Available**

File 4 : **File Not Available**

Search Notes



Search(Title or Subject)

Search results for 'hindi' ×

#	Subject	Notes Title	Creation Date	Action
1	Hindi	Hindi Vocabulary	May 23, 2024, 3:10 p.m.	Edit Delete

CONCLUSION

The project titled as **Online Notes Sharing System** was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time.

Computer has got clear advantage over the manual system. The computerized system is more reliable, efficient and fast at the end of the project, I can say that computer play a very crucial role in the development of firm. All the daily reports generated by the system are to be checked by the concerned official so as to ensure that all the transactions have been put through in appropriate accounts and this is tallied with the new vouchers. Computer does maximum work with in minimum time. Because it is used in every field so that it provides comfort and suitability to everyone. Providing maximum facilities and comfort to customers to customers is main goal of the firm. To achieve this goal, other modern facilities relating to computer should have to be provided.

BIBILIOGRAPHY

For Python and Django

[1] Django homepage. http://www.djangoproject.com/.

[2] Python documentation. http://www.python.org/doc.

[3] Django(web framework). http://en.wikipedia.org/wiki/Django.

[4] Django documentation. http://docs.djangoproject.com.

[5] Python(programming language). http://en.wikipedia.org/wiki/Python.

[6]Books: Web Development with Django by Ben Shaw, Saurabh,
Django 4 By Example by Antonio Mele

For MySQL

- https://www.mysql.com/
- http://www.mysqltutorial.org

For XAMPP

- https://www.apachefriends.org/download.html