Project on
Income Tax Management System

Submitted by
Team number-12 – 19BCE0646 – Abhishek S Nurandayyanmath
Team number-12– 19BCE0653 – Abhishek Sharma

School of Computer Science and Engineering,
Vellore Institute of Technology,
Vellore, Tamilnadu, India - 632014

I - Help file

Software used/required: 1) Front end: Notepad++: Editor for both backend and frontend code

2) Database: MySQL: Used for writing the backend code.

3) Server: XAMPP: Server to locally host the website.

4) Any other requirement: Web browser: For visual interaction and input from user.

Connection string used to connect the front end with database:

In index1.html(form):

```
<form action="backend_income.php" method="post" onsubmit="return validate()">
```

Backend with database:

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
// Create connection
$conn = new mysqli($servername,$username, $password);
// Check connection
if ($conn->connect_error) {
 die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
mysqli_select_db($conn,'income')
```

Hardware requirement: No special requirement.

Database details:

No. of tables as per the normalized schema : 17

No. of tables in the final project : 17

Front end details:

How many interface pages? : 7

Type of interface (web page/application) : Web Page + Form

How to install and test your application/web page?

Steps to install XAMPP on Windows:

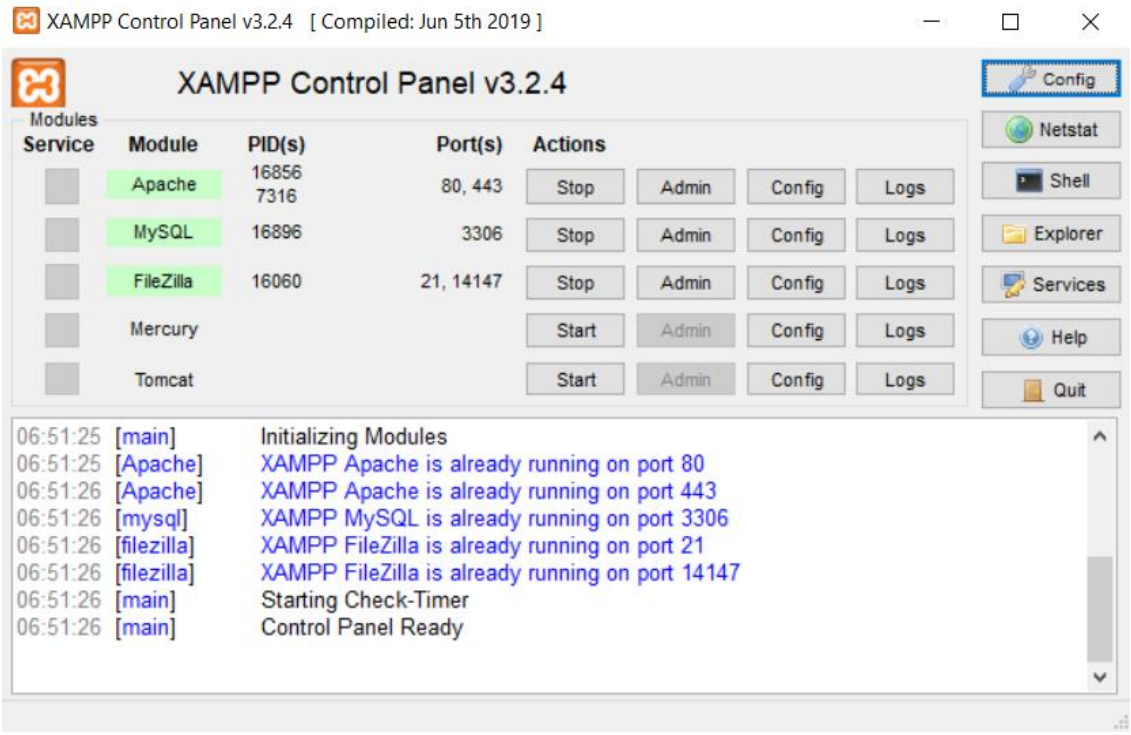
- In the web browser, visit Apache Friends and download XAMPP installer.

- During the installation process, select the required components like MySQL, FileZilla ftp server, PHP, phpMyAdmin or leave the default options and click the **Next** button.
- Uncheck the **Learn more about bitnami** option and click **Next** button.
- Choose the root directory path to set up the *htdocs* folder for our application. For example *'C:\xampp'*.
- Click the **Allow access** button to allow the XAMPP modules from the Windows firewall.

Using Xampp:

- After the installation process, click the **Finish** button of the XAMPP Setup wizard.
- Now the XAMPP icon is clearly visible on the right side of start menu. Show or Hide can be set by using the control panel by clicking on the icon.
- To start Apache,MySql and filezilaa just click on the **Start** button on the control panel.
- In XAMPP, the configuration files of Apache, MySQL, PHP are located in **C:\Program Files\xampp.** For any configuration file changes, you need to restart Apache and MySQL

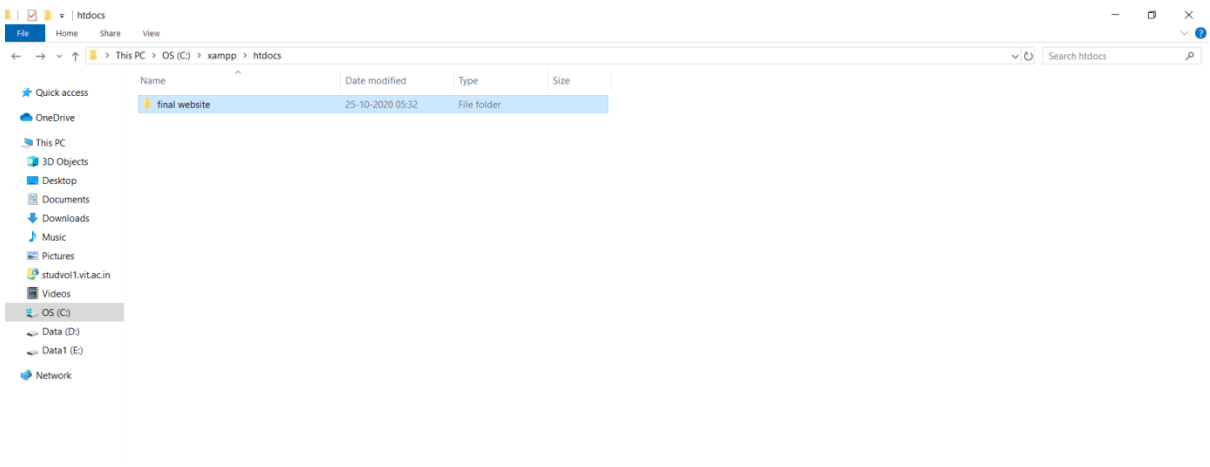
XAMPP control Panel:



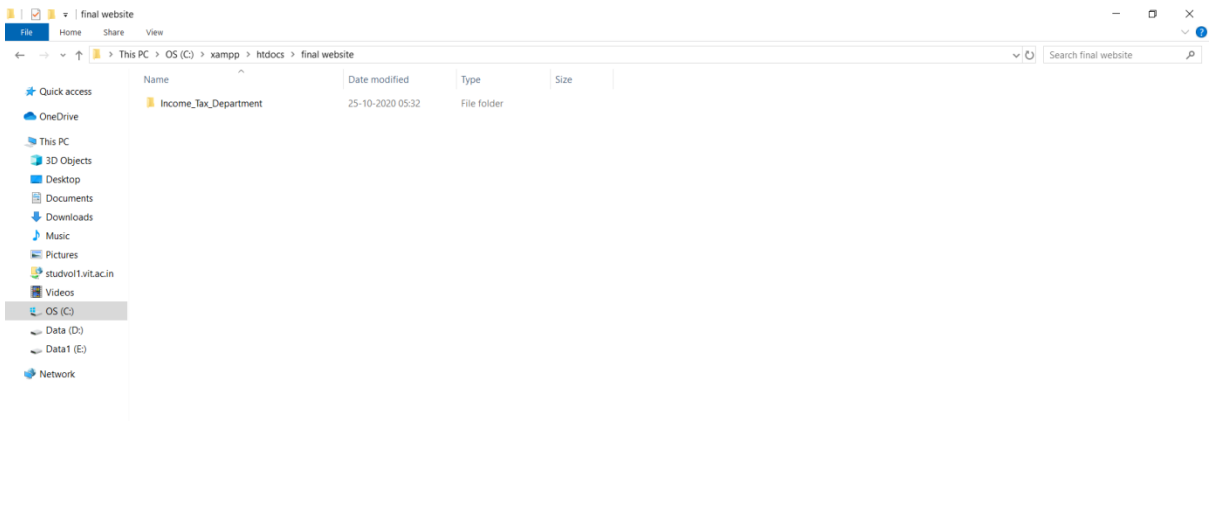
Placing the files in the right places:

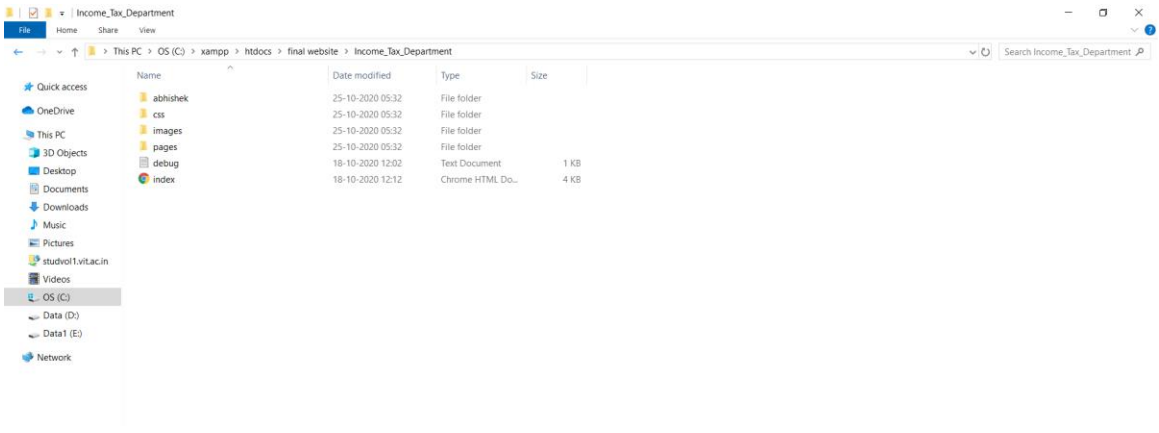
<u>For the website:</u>

- Unzip the zipped file provided in the google form link
- Navigate to Xampp folder in the location where you installed your application.
- Go to htdocs. This is where the “final website” folder must be pasted.



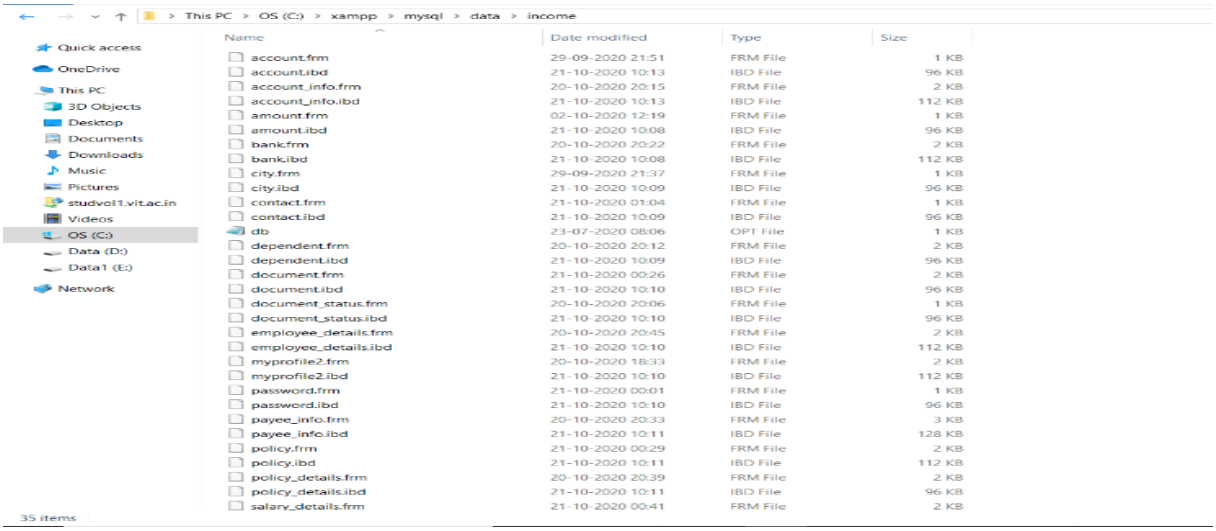
The contents of the folder final website should be:





For the database:

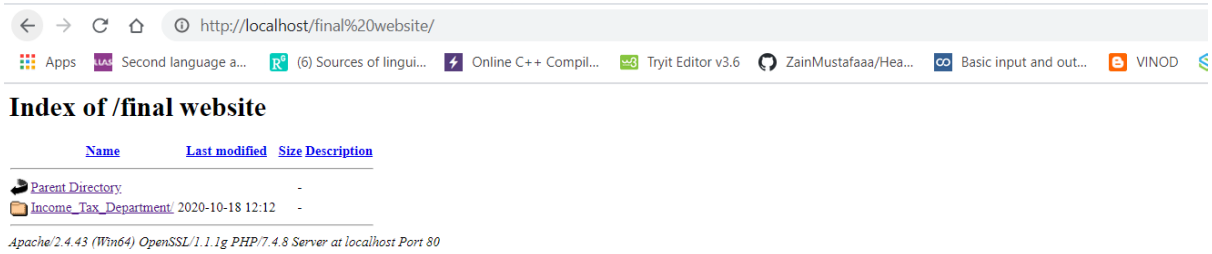
- Navigate to MySQL folder of Xampp.
- Click on the data folder and paste the folder “Income” attached in the google form.



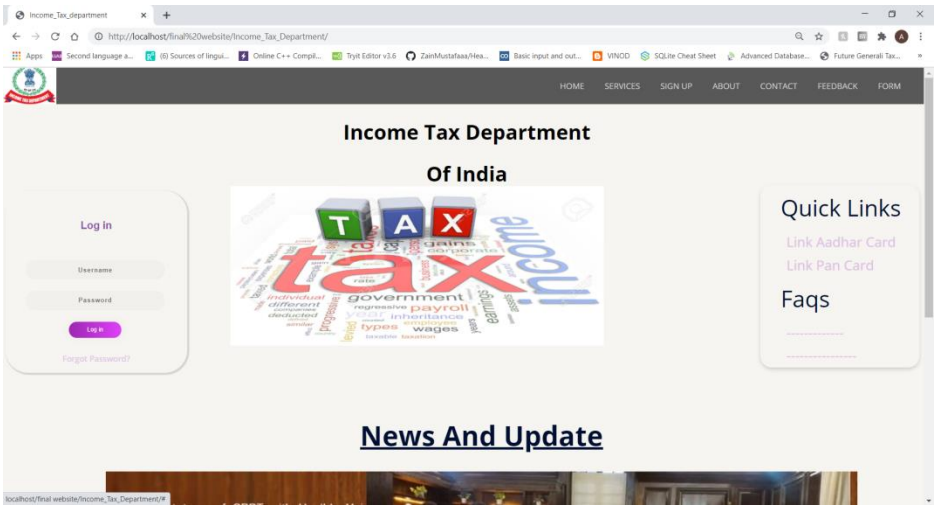
Once all the above steps are followed, our project is ready to be hosted locally on Xampp.

Hosting the website:

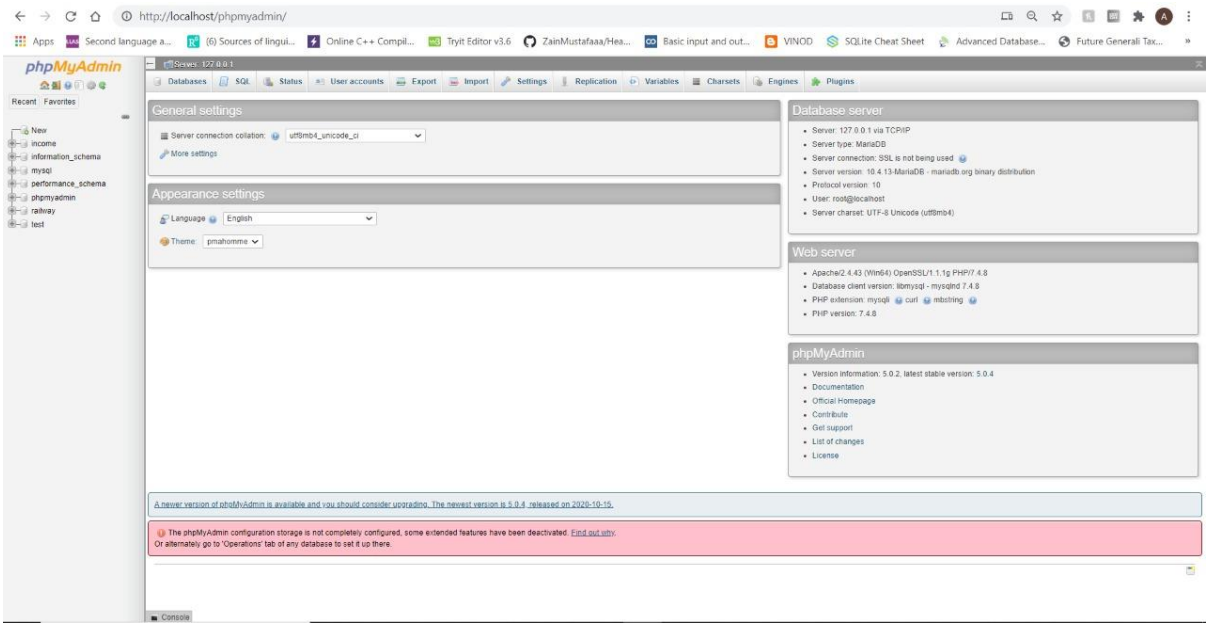
- Use the mentioned link to host the website: http://localhost/final%20website/.
- Click on “Income_Tax_Department/” and the website will be visible to you.



Website:



To see the database follow the link: http://localhost/phpmyadmin/


http://localhost/phpmyadmin/
phpMyAdmin
Recent Favorites
New
income
information_schema
mysql
performance_schema
phpmyadmin
railway
test
Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins
General settings
Server connection collation: utf8mb4_unicode_ci
More settings
Appearance settings
Language English
Theme: pmahomme
Database server
Server: 127.0.0.1 via TCP/IP
Server type: MariaDB
Server connection: SSL is not being used
Server version: 10.4.13-MariaDB - mariadb.org binary distribution
Protocol version: 10
User: root@localhost
Server charset: UTF-8 Unicode (utf8mb4)
Web server
Apache/2.4.43 (Win64) OpenSSL/1.1.1g PHP/7.4.8
Database client version: libmysql - mysqlnd 7.4.8
PHP extension: mysqli curl mbstring
PHP version: 7.4.8
phpMyAdmin
Version information: 5.0.2, latest stable version: 5.0.4
Documentation
Official Homepage
Contribute
Get support
List of changes
License
A newer version of phpMyAdmin is available and you should consider upgrading. The newest version is 5.0.4, released on 2020-10-15.
The phpMyAdmin configuration storage is not completely configured, some extended features have been deactivated. Find out why.
Or alternately go to 'Operations' tab of any database to set it up there.
Console

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our project guide Prof. Saravanakumar K for his exemplary guidance, monitoring and constant encouragement throughout the course of this subject CSE2004: Database Management Systems that helped us to complete this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the management of VIT UNIVERSITY for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

Lastly, we thank the almighty, our parents, brothers, sisters and friends for their constant encouragement without which this project would not be possible.

II – Phase I report

AIM:

This task has been undertaken by the students of B. Tech Computer science and engineering for the course of "Database Management Systems" with course code CSE2004. The essential methodology of this undertaking is to make a database for Income Tax Return System and interface it to a site with the goal that constant refreshing and exchanges on database can be taken care of.

The key purpose of our Income Tax Management System is to set up Tax Returns of a client. In Tax Information Framework System, a client registers himself enters all the nuances and moves various Documents that are significant for course of action of Tax Summary and Schedules for a gathering after productive convenience of the impressive number of reports.

PROBLEM STATEMENT:

The essential point of this undertaking is to set up a Tax rundown or Tax Returns framework for a customer. In annual expense forms framework, a customer registers himself at that point enters all the subtleties and transfers different Documents that are extremely vital for planning of Tax Summary.

PROPOSED SYSTEM OVERVIWEW:

In the upcoming months of the semester, we will try to extend our project to encompass the following goals:

1. **Login/register**
    - Clients should login to our website to update then profiles and start filling the form.
    - Administrator view will be provided through phpMyAdmin wherein the administrator can view the entire details of the applicant.

2. **Documents**
    - Specify any other statements relating the interest, divide or stocks.

3. **Accepting income tax registration form**
    - A basic form will be used to avail all the information regarding the person which is necessary for income tax filing.

4. **Payment**
    - Pay the tax registration fee through our website.

In this phase we have covered 3 steps of our database design which includes the following:

- **Deciding the functional and non-functional requirements of the user.**
- **Designing the ER diagram.**
- **Converting the ER diagram to conceptual schema.**

For the upcoming Phases we shall proceed further for Normalization and Physical design and implementation for the completion of our Database.

HARDWARE AND SOFTWARE REQUIREMENTS

This task for the ,most part, centers around the customer perspective on the database. This module will comprise of a landing page which will ask the client to Login (in the event that he as of now has enlisted). Else, he will be approached to "Sign in" with our site. Hence the site page will guide the client to the Income charge enlistment structure where he/she will be approached to enter his/her subtleties and announcements.

Hardware Requirements:

- Proper internet connection
- Laptop with enough space to accommodate a database.

Software Requirements:

- MySQL : Used for writing the backend code.
- Notepad++: Editor for both backend and frontend code
- Web browser: For visual interaction and input from user.
- XAMPP : Server to locally host the website.

DESIGNING THE CONCEPTUAL SCHEMA:

When all the necessities have been gathered and broke down, the following stage is to make an applied outline for the database, utilizing a significant level calculated information model.
The consequence of this stage is an Entity-Relationship (ER) chart. It is a progressed - level information model of use region. It gives a distinctive portrayal on how various elements or entities are identified with one another. It likewise portrays what qualities every entity set has.

Step 1: Identifying possible entities:

I. TAX REGISTRATION
II. DOCUMENT
III. DEPENDENTS
IV. BANK
V. PAYMENT
VI. STATUS
VII. MY_PROFILE
VIII. EMP_DETAILS
IX. POLICIES
X. TAX_BENEFIT

Step 2: Determining the appropriate matching attributes of each entity.

I. TAX REGISTRATION:

- Tax_no
- Name- composite attribute. The components of this attribute will be fname,mname and lname.
- sex
- contact_no
- User_id
- Password

II. DOCUMENT:

- Doc_id
- Doc_type
- Status:The values that will be accepted by this attribute are “accepted” or “rejected”

III. DEPENDENT:

- Tax_no-Weak entity with owner type TAX REGISTRATION
- Name: Composite attribute. The components associated with this attribute will be fname, mname and lname.
- Relation
- DOB

IV. PAYMENT:

- Tax_no
- Trans_id
- B_name
- Amount
- Date

V. BANK ACCOUNT:

- B_name
- Acc_Type
- taxid
- IFSCcode
- Acc_no

VI. STATUS:

- Tax_no
- Reg_status
- Sdate
- Status_time

VII. MYPROFILE:

- Tax_no
- Name: composite attribute. The components associated with this attribute will be fname, mname,lname.
- sex
- DOB
- Address: composite attribute. The components associated with this attribute will be city, street, state.
- Contact_no

VIII. POLICY

- P_no
- P_type
- iss_comp
- p_name
- exp_date
- iss_date
- tax_no
- p_term:derived attribute

IX. TAX_BENEFIT

This is a weak entity set and is strongly dependent on POLICY

- tax_no
- p_no
- tax_bene

X. EMPLOYEE

- Emp_name:composite attribute:this attribute will hold the fname,mname,lname.
- Emp_id
- Gr_sal
- Tax_no
- PAN

Step 3: Fixing the relationships between the various entities in our project.

i. MYPROFILE (a citizen) **registers for** TAX REGISTRATION.(1 to 1 relationship)

ii. MYPROFILE (a citizen) **Pays** PAYMENT.(1 to N relationship)

iii. MYPROFILE (a citizen) **holds** BANKACCOUNT(1 to N relationship)

iv. MYPROFILE (a citizen) **has** DEPENDENTS(1 to N relationship)

v. MYPROFILE (a citizen) **uploads** DOCUMENTS(1 to N relationship)

vi. STATUS of TAX REGISTRATION(1 to 1 relationship)

vii. MYPROFILE **updates** his/her policy.(1 to N relationship)

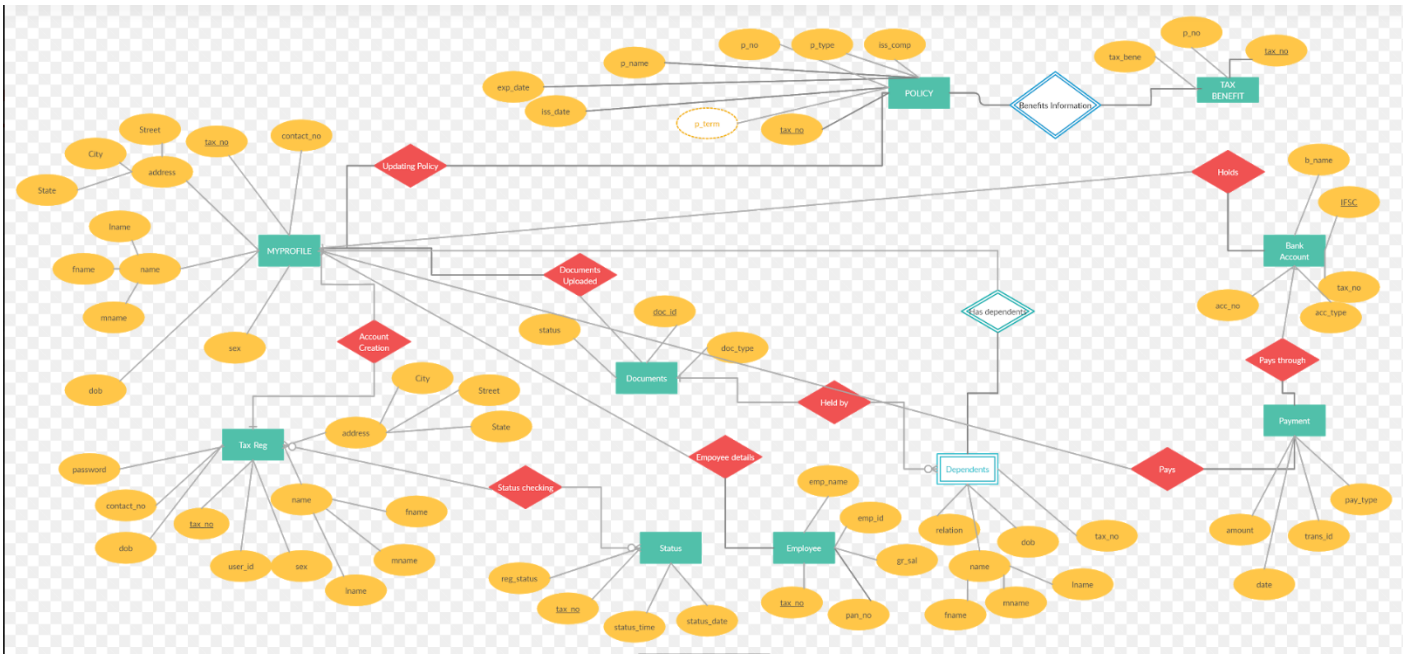
viii. MYPROFILE **works** at EMPLOYEE (1 to N relationship)

ix. PAYMENT **through** BANKACCOUNT.(N to 1 relationship)

x. DEPENDENTS **hold** DOCUMENTS.(1 to N relationship)

xi. POLICY **provides/gives** TAX_BENEFIT(1 to 1 relationship)

ER-DIAGRAM:


p_no
p_type
iss_comp
p_name
exp_date
iss_date
p_term
tax_no
POLICY
Benefits Information
tax_bene
p_no
tax_no
TAX BENEFIT
Updating Policy
Holds
b_name
IFSC
Bank Account
tax_no
acc_no
acc_type
Pays through
Payment
pay_type
amount
trans_id
date
Pays
Has dependents
Dependents
relation
dob
tax_no
name
fname
mname
lname
Held by
Documents
doc_id
doc_type
status
Documents Uploaded
Employee details
Employee
emp_name
emp_id
gr_sal
pan_no
Status checking
Status
reg_status
status_time
status_date
MYPROFILE
Street
City
address
State
tax_no
contact_no
lname
fname
name
mname
sex
dob
Account Creation
Tax Reg
password
contact_no
tax_no
dob
user_id
sex
name
fname
mname
lname
address
City
Street
State

REDUCING THE ER DIAGRAM TO CONCEPTUAL SCHEMA:

The result of the logical design phase is a set of relation schema. The ER diagram or class diagram is the foundation for these relation schemas. The relation schemas are the basis for table definitions. There are rules how the ER model or class diagram is transferred to relation schemas.
The following are the rules that we have kept in consideration while designing the conceptual schema of our project:

- **Strong Entity sets becomes a table.**
- **All single-valued attribute becomes a column for the table.**
- **A key attribute of the entity type represented by the primary key.**
- **The multivalued attribute is represented by a separate table.**
- **Composite attribute represented by components.**
- **Derived attributes are not considered in the table.**

CONCEPTUAL SCHEMA:

- MYPROFILE (tax_no, doc_id,dob,sex, street,state,city, fname, mname, lname)
  →The 'myprofile' entity set is a strong entity set where tax_no is the primary key
- DOCUMENT(docid,dtype,status)
- DOCUMENT_STATUS(docid,status,tax_no)
- TAX_REGISTRATION(tax_no,user_id,lname,fname,mname,password)
- CONTACT(tax_no,phone)
- DEPENDENT(tax_no,fname,mname,lname,dob,relation)
  →The 'dependent' entity set is a weak entity set because it strongly depends on 'Tax Registration'. This is the reason why we have added tax_no as a foreign key in this table
- REG_STATUS(tax_no,status,s_confirmation_date)
- PAYMENT(trans_id, tax_no,b_name,acc_no,amount,date)
- BANK_ACCOUNT(acc_no ,ifsc,b_name,tax_no, ,acc_type)
- POLICY(p_no ,tax_no,iss_comp,p_type,p_name,exp_date,iss_date)
- EMPLOYEE(emp_id,emp_name,gr_sal,pan_no,tax_no)

III – Phase II report

NORMALIZING THE RELATIONAL SCHEMAS:

Here we are going to demonstrate the normalization process for table MYPROFILE:

MYPROFILE:

Tax_no	Doc_id	DOB	Fname	Mname	Lname	Sex	Street	State	City
T_0001	D_0001	11-AUG-1995	Abhishek	Rajeev	Sharma	M	enclave	Maharashtra	Thane
T_0002	D_0002	15-AUG-1992	Kaustubh	Shankar	Shrivastava	M	seawoods	Maharashtra	Mumbai
T_0003	D_0003	08-MAY-1990	Ritwik	Arvind	Sinha	M	Powai view	Maharashtra	Mumbai
T_0004	D_0004	19-JUL-1991	Alia	deepak	Patel	M	Chitoor_road	Gujarat	Surat
T_0005	D_0005	11-AUG-1995	Akshat	Rajeev	aggarwal	M	enclave	Tamil_Nadu	Vellore
T_0001	D_0006	11-AUG-1995	Abhishek	Rajeev	Sharma	M	enclave	Maharashtra	Thane

Functional dependencies:

The functional dependencies associated with the above table are:

Tax_no →DOB Fname Mname Lname Sex Street State City -----(1)
Doc_id→DOB Fname Mname Lname Sex Street State City ------(2)
Doc_id→Tax_no--------(3)

State→City--------(4)

1NF

In Relational Database Management System (RDBMS), 1NF is one of the properties a relation (table) must satisfy. A table is said to be in 1NF if it satisfies the following properties:

Property 1: Any attribute (column) of a table must have been designed to accept only atomic values.
Property 2: Any value stored in any column must have single value (no repeating values).

The above table is in 1NF as it satisfies the above two condition i.e the columns can have atomic domains only and each column of the table must not have multiple values.

2NF

From functional dependency (1) and (2) we conclude that The candidate key for the relation is {tax_no,doc_id}

Candiadate key→ {tax_no,doc_id}

For a table to be in 2NF, it has to satisfy the following set of properties;

Property 1: The table should satisfy all the properties of previous normal form, i.e., 1NF. In other words, the table should be in 1NF.

Property 2: There should not be any partial key dependencies. This property is not applicable for relations (tables) which have single simple attribute as Primary Key. Hence, property 2 is applicable for relations those have more than one attribute combination as Primary Key (i.e., Composite Key).

This is because, there is the possibility of some of the attributes of the table to depend on any one or all of the attributes of the composite primary key.

The above table is in 2NF as it satisfies the above two condition i.e the table is in 1NF and it has no partial key dependencies i.e all the non-prime attributes of the table are completely dependent on the candidate key.

No decomposition required.

3NF

The following are the properties to be satisfied by a table for 3NF;

Property 1: The table should be in 2NF
Property 2: There should not be any Transitive Functional Dependency, i.e., there should not be any functional dependencies like a non-key (non-prime) attribute depends on another non-key (non-prime) attributes. Simply, we need all the non-key attributes must depend on the primary key only.

The above table is not in 3NF as the **FD: State→City** violates it.

We can convert the above table into 3NF by **decomposition**:

Schema after 3NF:

MYPROFILE(Tax_no,doc_id,DOB Fname Mname Lname Sex Street State City)

Tax_no	Doc_id	DOB	Fname	Mname	Lname	Sex	Street
T_0001	D_0001	11-AUG-1995	Abhishek	Rajeev	Sharma	M	enclave
T_0002	D_0002	15-AUG-1992	Kaustubh	Shankar	Shrivastava	M	seawoods
T_0003	D_0003	08-MAY-1990	Ritwik	Arvind	Sinha	M	Powai view
T_0004	D_0004	19-JUL-1991	Aaliy	deepak	Patel	M	Chitoor_road
T_0005	D_0005	11-AUG-1995	Akshat	Rajeev	aggarwal	M	enclave
T_0001	D_0006	11-AUG-1995	Abhishek	Rajeev	Sharma	M	enclave

CITY(State,City)

State	City
Maharashtra	Thane
Maharashtra	Mumbai
Maharashtra	Mumbai
Gujarat	Surat
Tamil_Nadu	Vellore
Maharashtra	Thane

BCNF

The rule of BCNF says that:
If we have set of FDs in R such that X → Y, then X must be a super key. In other words, if X is not a key, then the relation R is not in BCNF.
Since the City table has only two attributes associated with it so we don't need to check the BCNF conditions for it. Also MYPROFILE has {tax_id,Doc_id} as a key (a super key) and determines all the attributes uniquely. **Thus MYPROFILE and CITY are in BCNF.**

No decomposition of schemas required.

Final schemas after normalization:

MYPROFILE(Tax_no,doc_id,DOB Fname Mname Lname Sex Street State City)

Tax_no	Doc_id	DOB	Fname	Mname	Lname	Sex	Street
T_0001	D_0001	11-AUG-1995	Abhishek	Rajeev	Sharma	M	enclave
T_0002	D_0002	15-AUG-1992	Kaustubh	Shankar	Shrivastava	M	seawoods
T_0003	D_0003	08-MAY-1990	Ritwik	Arvind	Sinha	M	Powai view
T_0004	D_0004	19-JUL-1991	Aaliy	deepak	Patel	M	Chitoor_road
T_0005	D_0005	11-AUG-1995	Akshat	Rajeev	aggarwal	M	enclave
T_0001	D_0006	11-AUG-1995	Abhishek	Rajeev	Sharma	M	enclave

CITY(State,City)

State	City
Maharashtra	Thane
Maharashtra	Mumbai
Maharashtra	Mumbai
Gujarat	Surat
Tamil_Nadu	Vellore
Maharashtra	Thane

Final schemas of all the tables after normalization

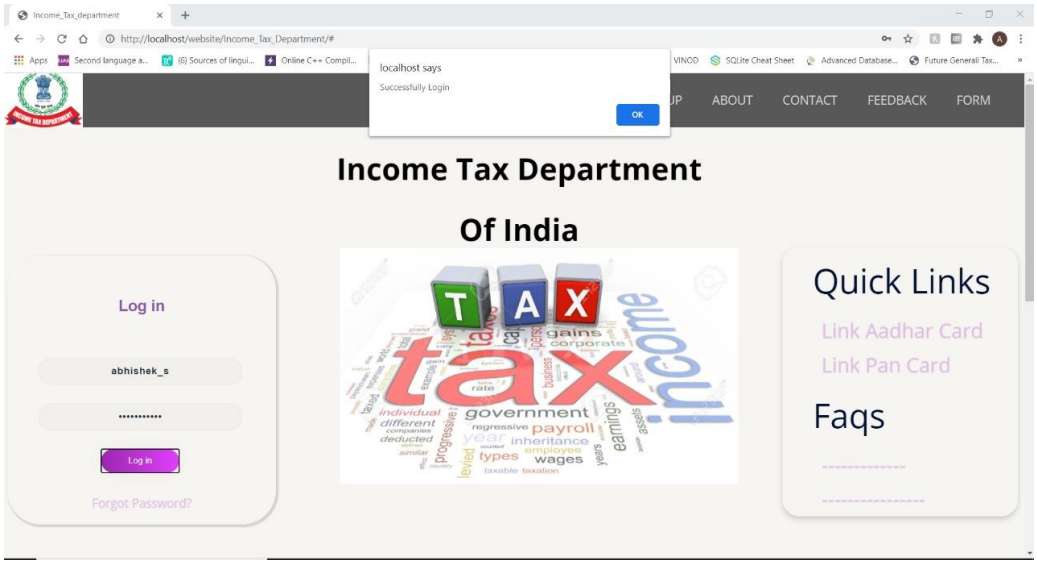
- MYPROFILE(Tax_no,doc_id,DOB Fname Mname Lname Sex Street State City)
- CITY(State,City)
- DOCUMENT(Docid,Dlife,Dtype)
- CONTACT(tax_no,phone)
- DOCUMENT_STATUS(docid,status,tax_no)
- TAX_REGISTRATION(tax_no,user_id,lname,fname,mname,password)
- DEPENDENT(tax_no,fname,mname,lname,dob,relation)
- REG_STATUS(tax_no,status,s_confirmation_date)
- ACCOUNT(Account_no,b_name)
- ACCOUNT_INFO(ifsc,account_no,type)
- BANK(Bankname,ifsc)
- PAYEE_INFO(Payment_id,Account_no,tax_id,b_name,Date)
- AMOUNT(b_name,amount)
- POLICY(tax_no, p_no)
- POLICY_DETAILS (p_no,p_type,p_name,iss_date,exp_date,iss_comp)
- SALARY_DETAILS(emp_id,emp_name,gr_sal)
- EMPLOYEE_DETAILS(emp_id,pan_no,tax_no)

IV – Phase III report

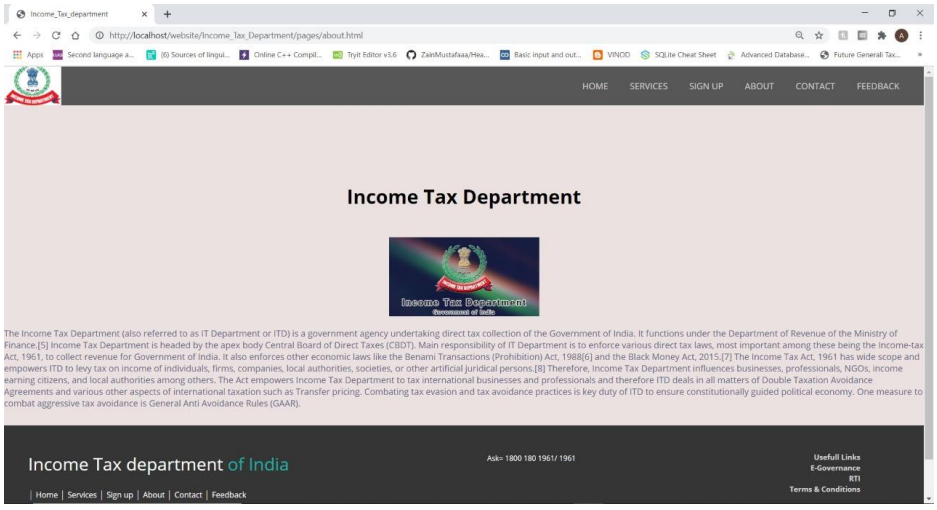
FLOW OF CONTROL AND PURPOSE OF INTERFACES

To create our website we have made use of HTML and CSS. PHP has been used to connect the database with the frontend (website/form). We have the following pages on our website:

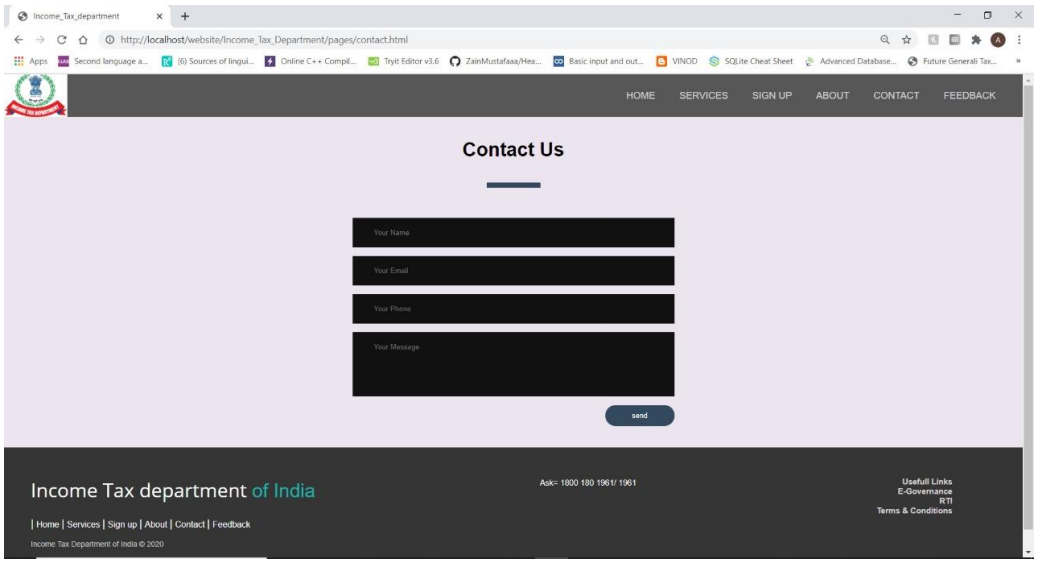
1) **Home:** This is the opening page of our website from where we can navigate to all other tabs.



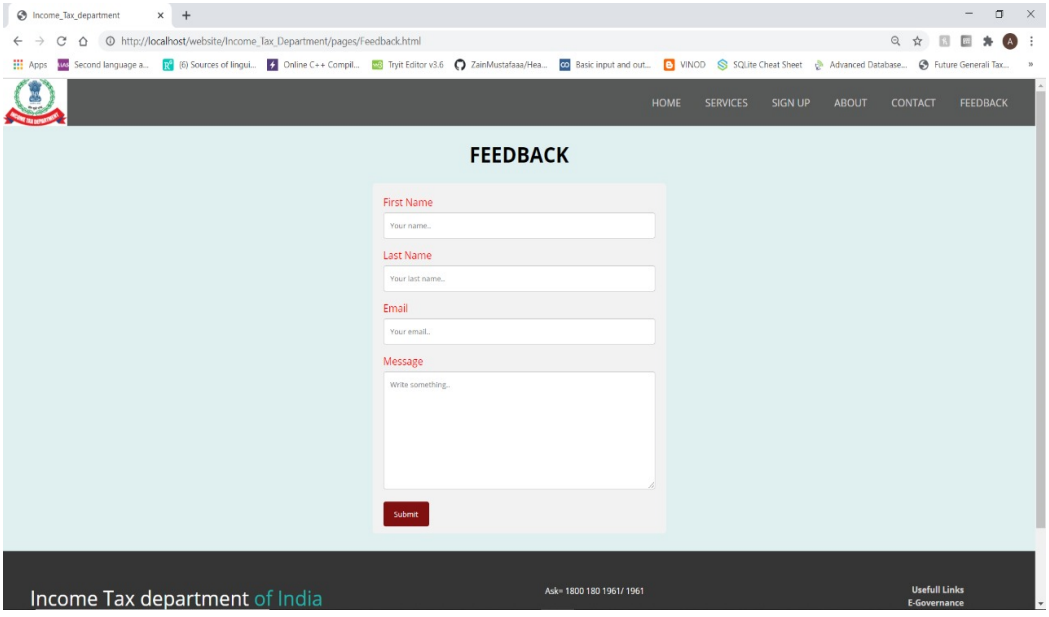
2) **About:** Displays general information about Income tax Department.



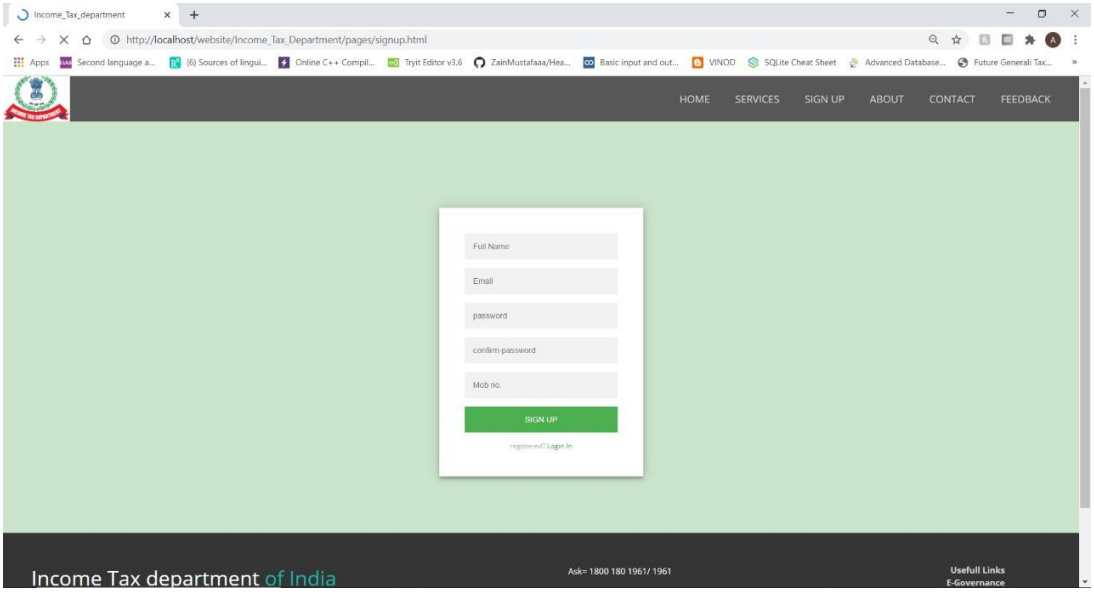
3) **Contact Us:** To help payee's resolve their queries.



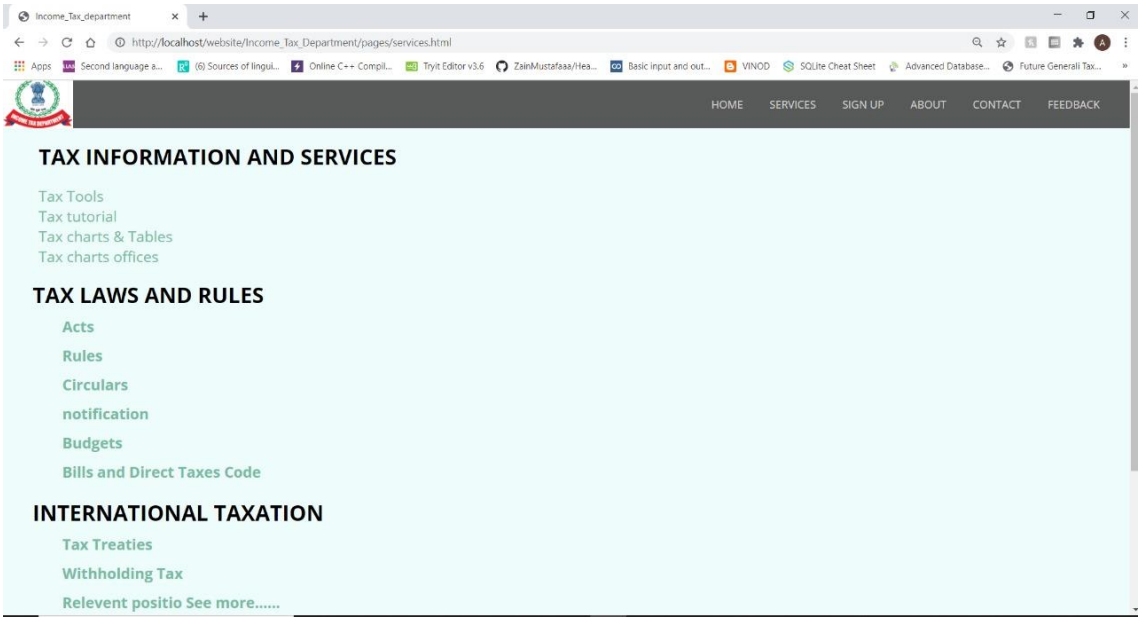
4) **Feedback:** We want to constantly improve our Income tax system, so this tab will help us take feedback from our users.



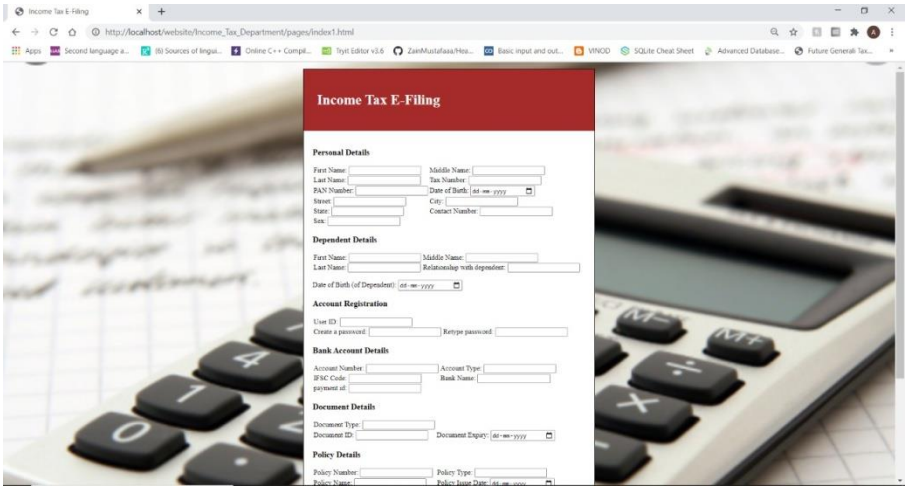
5) **Sign Up:** To allow the user to sign up an account.

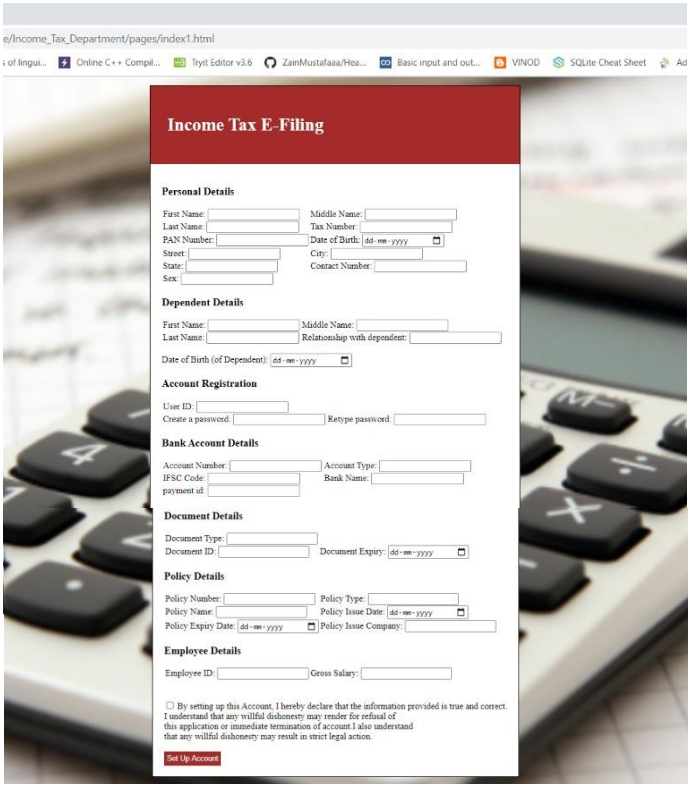


6) **Services:** This tab will hold the links of various services that are provided by the Income Tax Department of India.



7) **Form:** This is the main part of our project to input the information from the user. This has been designed in a user friendly way so that our users can conveniently use our website to register for their tax filing. This form has fields where the user can enter details. These details will be securely stored in the database.



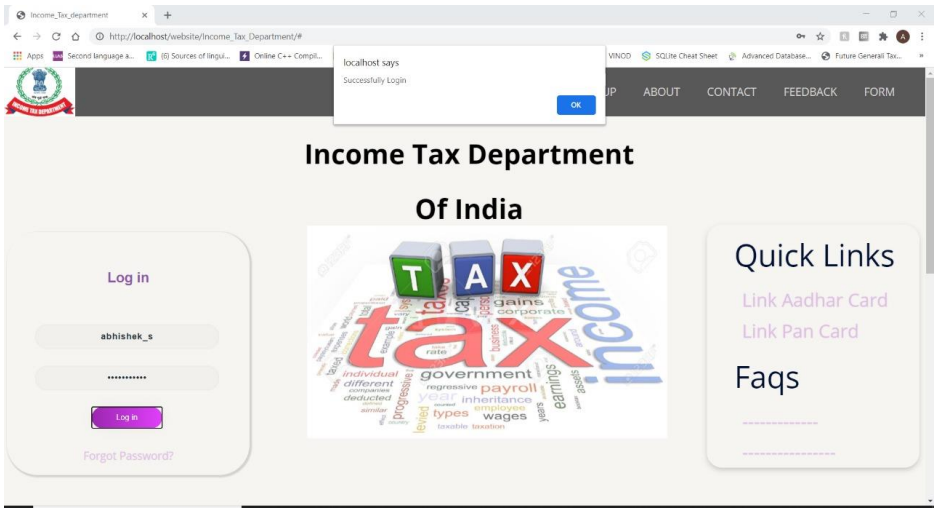


We also have a set of constraints so that we get the correct information from our users.

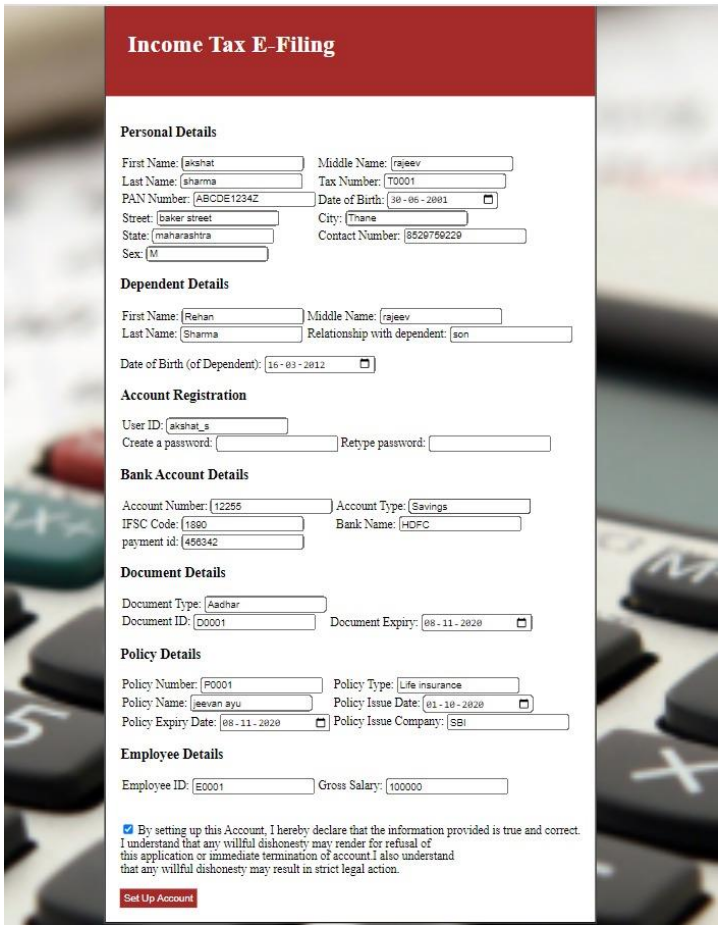
1. Contact Number: Length should be of 10 digits only.
2. PAN Number: Length should be of 10 characters only.
3. Tax ID: Length: 5 characters and should begin with ‘T’.
4. Policy no: Length: 5 characters and should begin with ‘P’.
5. Emp no: Length: 5 characters and should begin with ‘E’.
6. Docid: Length: 5 characters and should begin with ‘D’.

Working of our application

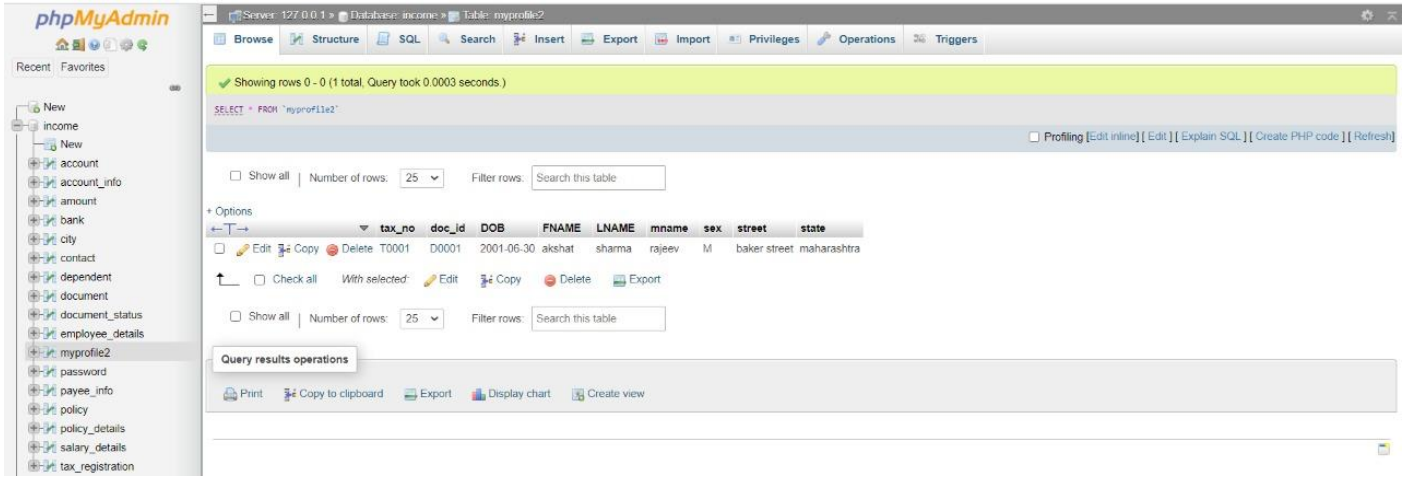
Step 1: Sign in using your credentials.

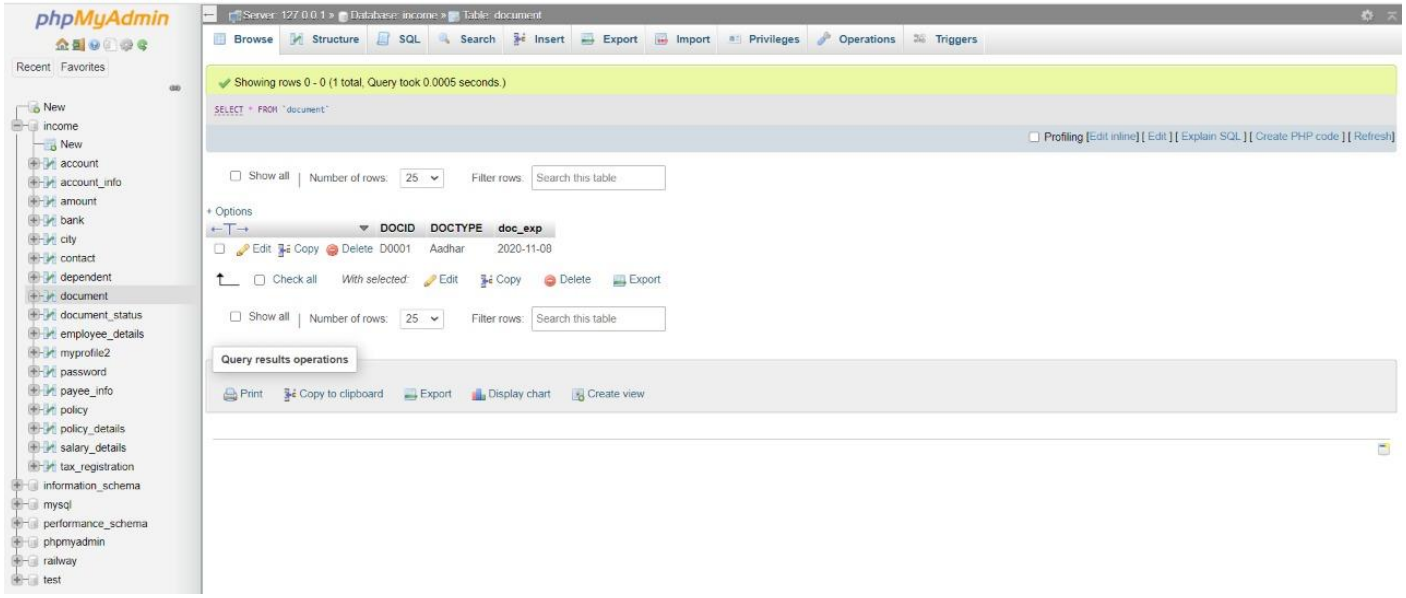


Step 2: Let us input some information as a user in the form.



Step 3: Values are stored in our database according to what the user has entered. Let us check two tables for validation if the values are properly stored or not.





We have successfully stored and registered our first user to our website.

LEARNING

Doing our project was like digging a tunnel of knowledge. We learnt many new things and how things actually work in real life. This project has helped us apply database concepts that in theory appear abstract but in real time system like in our case a tax registration system, hold utmost Importance. We realized the importance of planning the schema of our database using an ER diagram which helped us realize what the various entity sets were. It also helped us better comprehend the difference between strong entity sets and weak entity sets. Though concepts of normalization are taught well in class, the importance of normalizing tables is only understood when you work with them in real time systems. We learnt about HTML, CSS and a tad amount of JavaScript for front end web development. To implement out backend we learnt PHP, and using PHP we connected our form with the database we hosted in PhpMyAdmin. For database management, we understood SQL and used MySQL RDBMS. We also got to know about UI/UX designing. For that we used adobe XD. Overall, we are happy with the work we did and ample time we spent developing this project. We hope our efforts are realised by you as well.