

## 1. INTRODUCTION

### 1.1 OVERVIEW OF THE PROJECT

This project entitled as “Web Based Personal Savings Claim Process” has created in **Python** as front-end and **SQLite** as back-end. Now a days people invest their amount in various ways. The head of the family invest the amount unknowingly to that family members in various insurance companies, banks etc., the problem of the unknown investment is the amount which is saved for their family will not reach them at the proper time when the person meets death. These investments can reach their family member after a long time or it will not happen the amount will not useful for the family members.

This project gives the approach to solve this problem when the person meets a death in normal way or by an accident the death details will pass to the various insurance companies and banks. When the death details are registered so that necessary actions shall be taken by these companies thus the amount shall reach the corresponding family members in correct time.

The main objective is to make easy the process of claiming saving amount with effective way. It mainly focuses on citizens family can get their money after the saving person meet death. The proposed system is the online web application that helps the people to know about the savings of the citizens with efficient and secure manner. This web based saving claim process provides an online way for solving the problems faced by the public in turn saving the time.

This project contains four modules

- Citizen enrollment process
- Death certificate update
- Status update
- Claiming process

### 1.2 MODULES DESCRIPTION

There are four modules used in the project, the description of each module given below,

#### CITIZEN ENROLLMENT PROCESS

The citizen enrollment process module allows the citizens to authenticate in the website. People can register their personal details in a security manner. This personal information can't be accessed by unauthorized persons.

#### DEATH CERTIFICATE UPDATE

The user and admin both has access to this module. The user will update the details about the citizen death and also the aadhar id of the person. Now the admin can access the process with the help of aadhar id to know whether the citizen have any saving policies. This process is important for every citizen.

#### STATUS UPDATE

The admin has access to this module. When the death informations are given then the saving details of the citizen's status will be update. This personal information can't be accessed by unauthorized persons.

#### CLAIMING PROCESS

User can get the status information through mail or through personal information which registered. Where this process is helpful for the users to claim the savings of the citizen.

## 2. SYSTEM STUDY

### 2.1 EXISTING SYSTEM

Here the existing system is a manual one when the citizen can only know about his savings there is no way to claim the saved money after citizen's death.

The following are the reasons why the current system should be computerized:

- To increase efficiency with reduced cost.
- To reduce the burden of paper work.
- To make more user friendly interface.
- To generate required reports easily.
- To facilitate the services from online.

#### Limitations of existing system

- Time consumption
- Paper work
- Storage requirements
- Less reliable
- Accuracy
- Difficulty in keeping new records

To avoid all these limitations and make the system working more accurately it needs to be computerized

### 2.2 PROPOSED SYSTEM

Proposed to make the new system extremely user friendly the proposed system the entire saving claim process. The savings claiming process for the citizen's family is the main concept of the project. Proposed system is an online web application that helps people posts their problems in an efficient and secure manner.

## ADVANTAGE OF PROPOSED SYSTEM

- Using Python language it is more securable for records and details.
- It is user friendly language when we handled with the host as real time project.
- It maintains status for every user who registered to know about the savings.
- Manages member information separately for all exercise considering the requirements of this system.

## 2.3 FEASIBILITY ANALYSIS

Feasibility is the determination of whether or not a project is worth doing. The processes is followed in making this determination is called a feasibility study. Feasibility study is the test of system proposal according to its workability, Impact on the organization ability to meet user's needs, and effective use of resources. The result of feasibility study is a formal proposal. This is simply a report – a formal document detailing the nature and scope of the proposed solution.

### 2.3.1 TECHNICAL FEASIBILITY

A system that can be developed technically and that will be used if installed must still be a good invested for the organization. The assessment of technical feasibility must be based on an outline design on system requirements in terms of inputs, outputs, files, programs, procedures. Technical feasibility centers around the existing computer system and to what extend it can support the proposed system. The current technical resources, which are available in the organization, are capable of handling the requirements in the aspect of technical staff.

### 2.3.2 ECONOMICAL FEASIBILITY

The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. Economic feasibility deals with the analysis of costs against benefits (i.e.) whether the benefits to the enjoyed due to the new system are worthy when compared to the costs to be spent on the system. Economic analysis is the nose frequently used technique for evaluating the cost effectiveness of the proposed project. More commonly known as cost / benefit analysis, the procedure is to determine whether the project have the benefits and savings. Further compared with the existing-costs in the manual procedure, the current project involves less investment. The cost when compared to the benefits of the system are much low. Hence the system is economically feasible.

### 2.3.3 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned into information systems that will meet the bank's operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. There are questions that will help to test the operational feasibility of a project. Is there sufficient support for the project from management? Have the users being involved in the planning and development of the project. Will the proposed system cause harm?

1. The changes brought to the system.
2. The operational skills that will be required for entering data and the training to be given are also considered.

## 3. SYSTEM SPECIFICATION

### 3.1 HARDWARE SPECIFICATION

Processor	:	INTEL(R) CORE(TM)i3
Speed	:	1.70 GHz
RAM	:	4.00 GB
System Type	:	64 Bit OS
Keyboard	:	104 Keys
Mouse	:	Scroll Mouse

### 3.2 SOFTWARE SPECIFICATION

Development Platform	:	Windows 10
Front-End Tool	:	Python using Django framework
Back-End Tool	:	SQLITE
Documentation	:	Microsoft Word

### 3.2.1 About the Front End

#### PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

#### FEATURES OF PYTHON PROGRAMMING:

- A simple language which is easier to learn: Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.
- Free and open-source: You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code. Python has a large community which improves in each iteration.
- Portability: You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.
- Extensible and Embeddable: Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.
- A high-level, interpreted language: Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.
- Large standard libraries to solve common tasks: Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQL dB library using import MySQL dB .Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.
- Object-oriented: Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating objects.

#### APPLICATIONS OF PYTHON:

- Web Applications: You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.
- Scientific and Numeric Computing: There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.
- Creating software Prototypes: Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.
- Good Language to Teach Programming: Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest languages to learn because of its simple easy-to-use syntax.

#### DJANGO

With Django, you can take Web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. This follows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF),

- Ridiculously fast: Django was designed to help developers take applications from concept to completion as quickly as possible.
- Fully loaded: Django includes dozens of extras you can use to handle common Web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.
- Reassuringly secure: Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and click jacking. Its user authentication system provides a secure way to manage user accounts and passwords.
- Exceedingly scalable: Some of the busiest sites on the planet use Django's ability to quickly and flexibly scale to meet the heaviest traffic demands.
- Incredibly versatile: Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

- Ports to other languages: Programmers have ported Django's template design from Python to other languages, providing decent cross-platform support. Some of these options are more direct ports; others, though inspired by Django and retaining its concepts, take the liberty to deviate from Django's design.

About the Django Software Foundation:

Development of Django is supported by an independent foundation established as a 501(c) (3) non-profit. Like most open-source foundations, the goal of the Django Software Foundation is to promote, support, and advance its open-source project: in our case, the Django Web framework.

The foundation's goals are to:

- Support development of Django by sponsoring sprints, meatus, gatherings and community events.
- Promote the use of Django among the world-wide Web development community.
- Protect the intellectual property and the framework's long-term viability.
- Advance the state of the art in Web development.

### Hyper Text Markup Language

HTML or Hyper Text Markup Language is a simple scripting language that is interpreted within a web browser. HTML helps to make web pages. HTML is the basis behind (almost) all web pages. Some tags in an HTML document determine the way certain text, such as titles, will be formatted. Other tags cue the computer to respond to the user's action on the keyboard or mouse. An important tag is a link, which may contain the Uniform Resource Locator (URL) of another document. The URL can be compared to an address where a particular document resides.

The document may be stored on the same computer as the parent document or on any computer connected to the World Wide Web. The user can navigate from document to document simply by clicking on these links. HTML also includes mark ups for forms, which let the user fill out information and electronically send, or email the data to the document author and initiate sophisticated searches of information on the internet, or good and services.

### BOOTSTRAP

Build responsive, mobile-first projects on the web with the world's most popular front-end component library. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

It supported by all popular browsers. With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Provides a clean and uniform solution for building an interface for developers.

### CSS

**Cascading Style Sheets (CSS)** is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate. css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same mark up page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

### MOST POPULAR PYTHON AND DJANGO WEBSITES

- Python
- YouTube
- Google
- DropBox
- Yahoo Maps
- Instagram

### BACK END

#### SQLITE

SQLite is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client-server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is

arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

SQLite Features:

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example, it partially provides triggers, and it cannot write to views (however it provides instead of triggers that provide this functionality). While it provides complex queries, it still has limited alter table function, as it cannot modify or delete columns.

SQLite uses an unusual type system for an SQL-compatible DBMS; instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is dynamically typed. Moreover, it is weakly typed in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the column's preferred type is integer). This adds flexibility to columns, especially when bound to a dynamically typed scripting language. However, the technique is not portable to other SQL products. A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products. The SQLite web site describes a "strict affinity" mode, but this feature has not yet been added. However, it can be implemented with constraints like CHECK (type of(x) ='integer').

SQLite with full Unicode function is optional. Several computer processes or threads may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an error code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on enabling concurrent reads and writes.

SQLite version 3.7.4 first saw the addition of the FTS4 (full text search) module, which features enhancements over the older FTS3 module. FTS4 allows users to perform full text searches on documents similar to how search engines search webpages. Version 3.8.2 added support for creating tables without rowid, which may provide space and performance improvements. Common table expressions support was added to SQLite in version 3.8.3.

In 2015, with the json1 extension and new subtype interfaces, SQLite version 3.9 introduced JSON content managing.

## 4. SYSTEM DESIGN

### 4.1 DATA FLOW DIAGRAM

The DFD takes an input-process-output view of a system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

Data objects represented by labeled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embodies the applications.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

#### RULES FOR DFD:

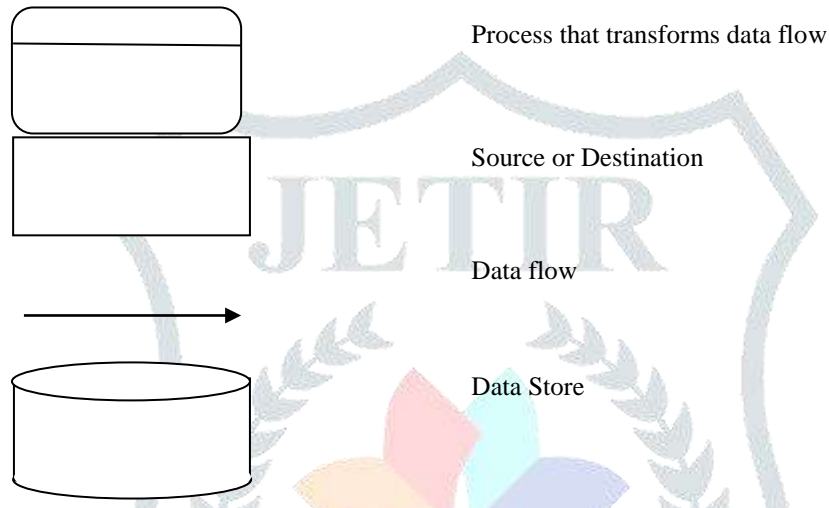
- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions
- Reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to the system with Rounded circles.
- A process is required for all the data transformation and Transfers. Therefore, never connect a data store to a data Source or the destinations or another data store with just a Data flow arrow.
- Do not indicate hardware and ignore control information.
- Make sure the names of the processes accurately convey everything the process is done.
- There must not be unnamed process.
- Indicate external sources and destinations of the data, with Squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except simple Record retrievals.

- Label data flow on each arrow.
- Use details flow on each arrow.
- Use the details flow arrow to indicate data movements.

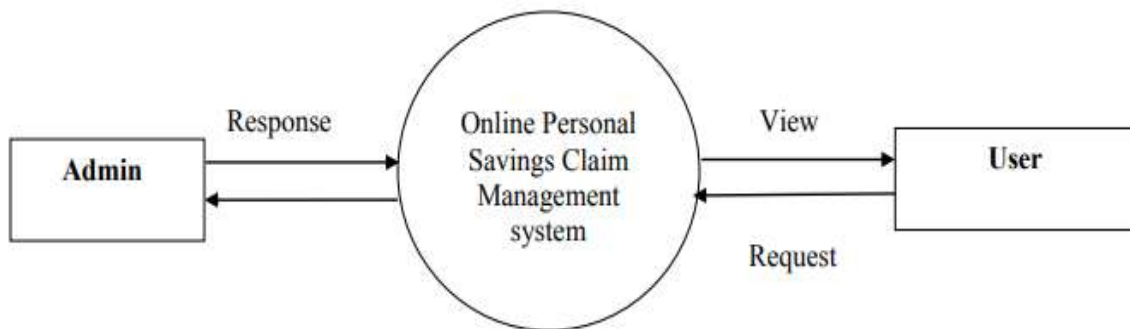
**DFD Symbols**

In the DFD, there are four symbols

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.

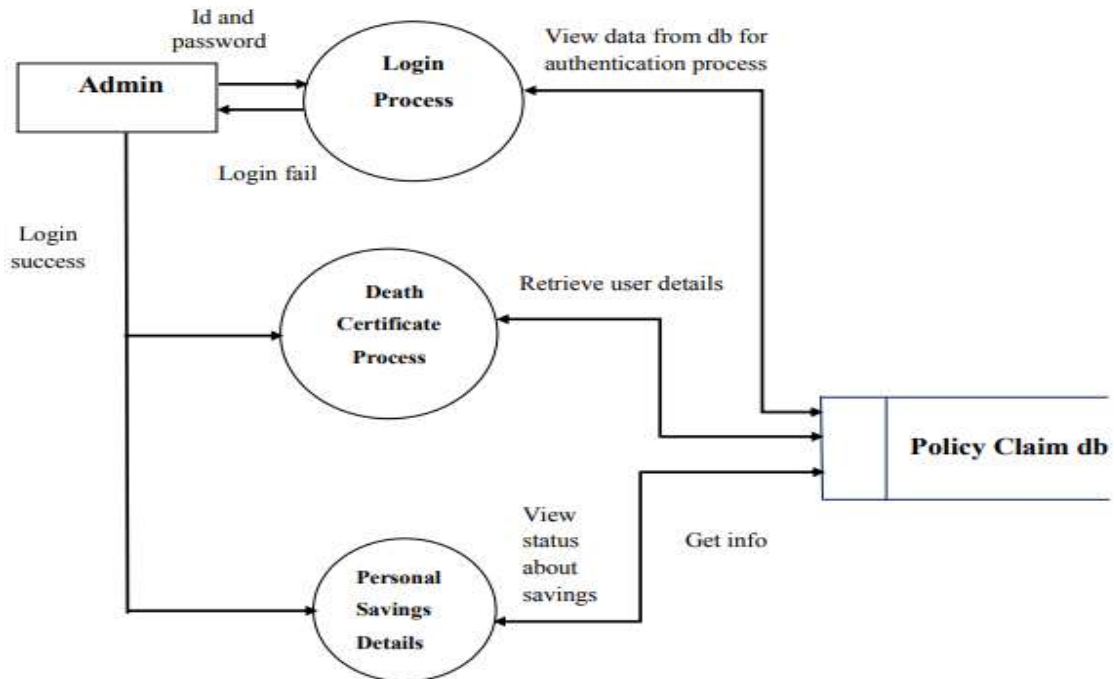


Level-0



LEVEL 1

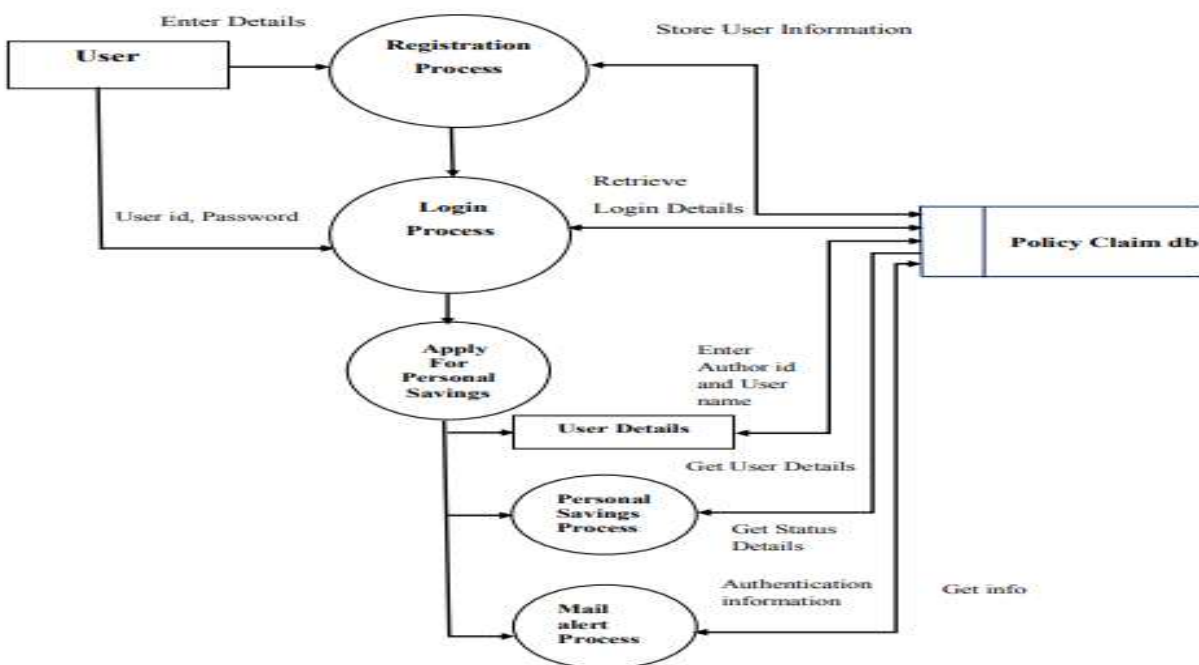
ADMIN PAGE



LEVEL 1



USER PAGE



#### 4.2 TABLE DESIGN

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users. During database design the following objectives are concerned:-

- Controlled Redundancy
- Data independence
- Accurate and integrating
- More information at low cost
- Recovery from failure
- Privacy and security
- Performance
- Ease of learning and use.

#### TABLE DESIGN

**TABLE NAME: CITIZEN TABLE**

**PRIMARY KEY: Id**

Field name	Description
Id	id of the user
First name	First name of the user
Last name	Last name of the user
Citizen_name	Name of the user
Aadhar_id	Aadhar number of the person
Mobile_number	Citizen Mobile number
Age	Age of the Citizen
Gender	Gender of the Citizen
Emailid	Email id of the Citizen
Address	Address of the Citizen
DOB	Date of birth information

TAB-1.a)

**TABLE NAME: SUPERUSER**

Field name	Description
Username	Admin's username for login
Password	Authentication password

TAB-1.b)



### 3.SAVINGS TABLE

PRIMARY KEY: Id

Field name	Description
Id	Id of the user
First name	First name of the user
Last name	Last name of the user
Citizen_name	Name of the citizen
Aadhar_id	Aadhar number of the citizen
Salary	Salary amount
Annual_income	Annual income of the citizen
Address	Address of the citizen
Current_savings	Current savings amount
Status	Status about the savings

TAB-1.c)

### 4.DEATH TABLE

PRIMARY KEY: Id

Field name	Description
Id	Id of the User
Dod	Death date of the citizen
Gender	Gender of the citizen
Name	Name of the citizen
Another	Relative of the death person
Pod	Place of death
Aadhar-id	Aadhar id of the citizen

TAB-1.d)

### 4.3 INPUT DESIGN

Input Design converts the user-oriented inputs to computer-based formats. Inaccurate input data are the most common cause of errors in data processing. Error data entered by the data operator can be controlled by the input design. The goal of designing input is to make the data entry easy, logical and as free from errors as much as possible.

The proposed system is completely menu-driven. It is a powerful tool for interactive design. It helps the user comprehend the range of alternatives available and also prevents them from making an invalid selection. All entry screens are interactive in nature. It has been designed taking into account all the constraints of the end-user.

#### REGISTRATION PAGE



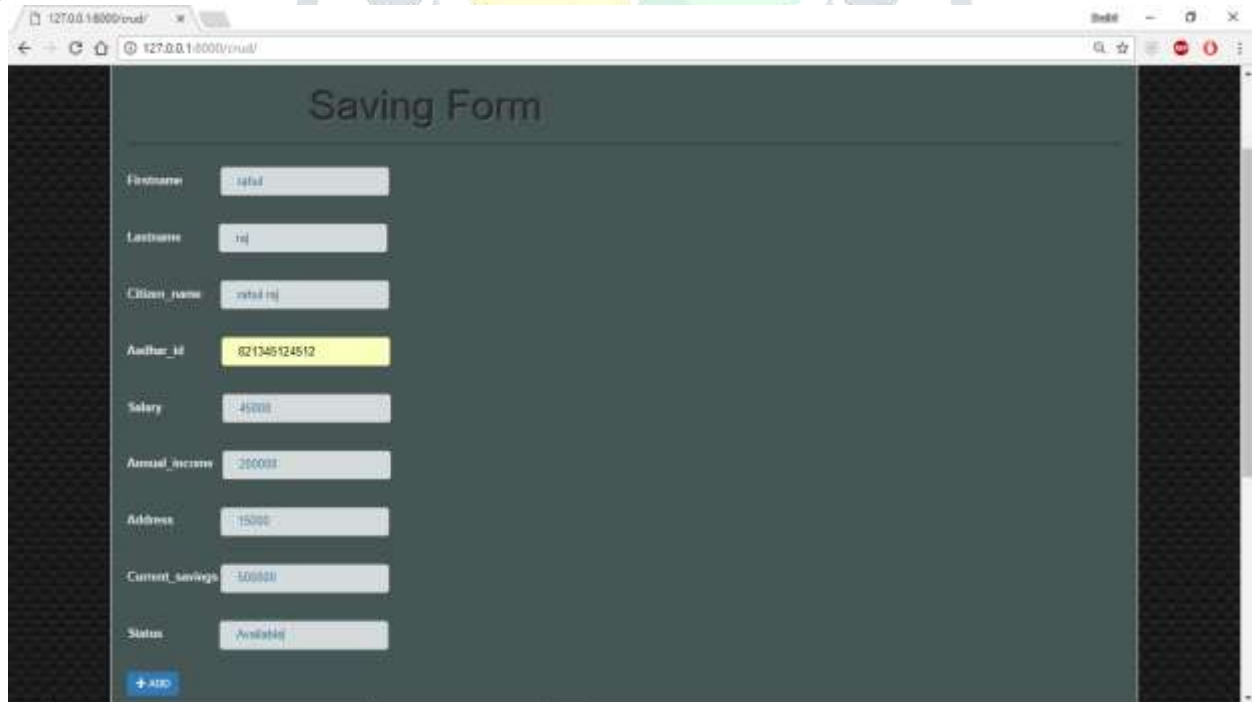
The screenshot shows a web form titled "Citizen Registration". The form contains the following fields and values:

Field	Value
Firstname	niket
Lastname	raj
Citizen_name	niketraj
Author_id	84574854125
Mobile_number	8945741874
Age	22
Gender	male
Email_id	niketraj@gmail.com
Address	12345
DOB	10-11-1996

A blue "+ADD" button is located at the bottom left of the form.

FIG-1.a)

#### CLAIM PAGE



The screenshot shows a web form titled "Saving Form". The form contains the following fields and values:

Field	Value
Firstname	niket
Lastname	raj
Citizen_name	niketraj
Author_id	821345124512
Salary	45000
Annual_increment	200000
Address	12345
Current_savings	500000
Status	Available

A blue "+ADD" button is located at the bottom left of the form.

FIG-1.b)

DEATH UPDATE



FIG-1.c)

4.4 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system. For many end-users, output is the main reason for developing the system and the basis on which they evaluate the usefulness of application. The objective of a system finds its shape in terms of the output. The analysis of the objective of a system leads to determination of outputs. Outputs of a system can take various forms. The most common are reports, screens displays, printed form, graphical drawing etc. the output also vary in terms of their contents, frequency, timing and format.

This section describes of the system output design relative to the user/operator; show a mapping to the high-level data flows .System outputs include reports, data display screens and GUIs, query results.

- Identification of codes and names for reports and data display screens
- Description of report and screen contents (provide a graphic representation of each layout and define all data elements associated with the layout or reference the data dictionary)
- Description of the purpose of the output, including identification of the primary users.

REGISTERED REPORT

Firstname	Lastname	Citizen_name	Aadhar_id	Mobile_number	Age	Gender	Email_id	Address	DOB	Action
boopathi	kumar	kumar boopathi	845267415874	9846578421	45	male	boopathi@gmail.com	coimbatore	20.06.1972	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
mohan	babaji	babaji mohan	845698754684	9875469875	22	male	babaji@gmail.com	selam	16.02.1995	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
rohan	babaji	rohan babaji	7658451257	9857884565	24	male	rohan@gmail.com	erode	02.06.1993	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
seema	raj	seema raj	754268451454	9874562214	23	female	seema@gmail.com	coimbatore	05.01.1994	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Guru	raj	guru raj	786548798822	9854655254	22	male	guru53@gmail.com	erode	06.07.1995	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
rahul	raj	rahul raj	845745654125	9845745874	22	male	rahul654@gmail.com	erode	10.12.1995	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

FIG-2.a)

## SAVINGS REPORT

Firstname	Lastname	Citizen_name	Aadhar_id	Salary	Annual_income	Address	Current_savings	Status	Action
boopathi	kumar	boopathi kumar	845745654125	300000	550000	erode	1500000	Available	 
rohan	balaji	rohan balaji	823146512346	25000	500000	ram nagar coimbatore	1000000	Available	 
mohan	balaji	mohan balaji	854123651243	25000	500000	balaji nagar,coimbatore	600000	Not Available	 
raahul	raj	raahul raj	821345124512	45000	200000	15000	500000	Available	 

FIG-2.b)

## CERTIFICATE REPORT









Date of death	Gender	Name	Father/ Husband	Place of death	Aadhar_id	Action
2018-03-02	male	mohan	chinnaiya	selam	845236541452	 
2018-01-25	female	Seema	pandiyar	coimbatore	854123651243	 
2017-12-18	male	yuvaraj	nagaraj	theni	823146512346	 
2017-10-20	male	naveen	salvam	Erode	821345124512	 

FIG-2.c)

## 5. SYSTEM TESTING AND IMPLEMENTATION

## 5.1 SYSTEM TESTING

System testing makes a logical assumption that if all the paths of the system are correct, the goal will be successfully achieved. System testing is aimed at ensuring that the system works accurately and efficiently before it can be operated on live data. The logical design is thoroughly done and continuously examined. Inadequate testing may lead to error & may explode into a much larger problem in course of time if not treated properly. The proper choice of test is an important System Testing makes a logic assumption that if all the paths of as test itself. The main objective for testing is described as follows:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as yet undiscovered error.
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Features to be tested
- Verify that the entries are of the correct format

- No duplicate entries should be allowed
- All links should take the user to the correct page.

Software is the only one element of a large computer-based system. Ultimately, software is incorporated with other system elements (e.g., Hardware, People, Information), and a series of system integration and validation tests are conducted. These tests fall outside the scope of the software process and are not conducted solely by software engineers. However, steps taken during software design and testing can greatly improve the probability of successful software integration in the larger system.

## TEST PLAN

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to the user. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

### Unit Testing

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. In the unit testing the entire project module is tested individually.

- Citizen enrollment process
- Death certificate update
- Status update
- Claiming process

TEST CASE	TEST CASE DESCRIPTION	EXPECTED RESULT	OBSERVED RESULT	RESULT PASS/FAIL
Enter the details for registration process.	Check whether all the fields are entered.	All fields are entered.	Every field has entered correctly.	Pass

### System Testing

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements.

System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

TEST CASE	TEST CASE DESCRIPTION	EXPECTED RESULT	OBSERVED RESULT	RESULT PASS/FALL
Testing all the modules in the system	It tests all the modules in the system whether it is executing in the system or not	By testing all the modules it is accepted by the system and should execute the result	It tests all the modules and the system accept all the modules and it produces an expected result	pass

### Integration Testing

In integration testing, the tested modules are combined into modules, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules.

The User enrollment process, **Grievance** process, Priority assignment, Grievance view/update, Mail alert process Complaint forward process details the entire module integer and tested effectively.

TEST CASE	TEST CASE DESCRIPTION	EXPECTED RESULT	OBSERVED RESULT	RESULT PASS/FALL
Testing all the modules to verify Combined Functionality after integration	It tests all the modules based on the input from the user to the system and getting an valid output from the system	Valid input from the user to the system and getting a valid output from the system	Getting a valid and reliable output from all the modules in the system	pass

### Acceptance Testing

The objective of the acceptance test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later.

### 5.2 SYSTEM IMPLEMENTATION

Implementation is the state in the project where theoretical design turned into working system. There are several methods for handling the implementation and consists for changing from the old to the new computerized system. The most secure method for conversion from the old system is to run the old and new system in parallel. In this approach; a person may operate in the manual processing system as well as start operating the new computerized system.

Another commonly used method is a direct cut over the existing manual system to the computerized system. The change may be within a week or a day. This strategy requires planning. A working version of the system can also be implemented in one part of the organization and the changes can be made as and when required, but this method is less preference due to the loss of entire system. After the system is implementation, a review should be conducted to determine whether the systems are meeting expecting where improvements are needed.

Implementation plan includes a description of all activities that must occur to implement the new system and to put into operation. It defines the personal responsible for the activities and prepares a time chat for implementation the system. The implementation plan should anticipate possible problems and must be able to deal with them. The usual problem may be missing documents, missed data formats between current and new files, errors in data translation, missing data etc.

### 6. CONCLUSION

The project "**WEB BASED PERSONAL SAVINGS CLAIM PROCESS**" was designed successfully and it was executed with successful database connectivity. Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. Proposed system is concluded that the application works well and satisfies the needs. The application is tested very well and errors are properly debugged. This project mainly focuses on development for web based personal savings claim process provides an online way for solving the problems faced by the public in turn saving the time.

### 7. SCOPE FOR FUTURE ENHANCEMENT

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. In this application we can include the future enhancement for develop in this application in android smart phone application. Now a day's peoples all are using the smart phone like android based phones so that would help to get the information easily through smart phone and get a response efficient manner.

### BIBLIOGRAPHY

The following books and websites were very helpful during the completion of project:

### REFERENCES

- [1] van Rossum, Guido (5 June 2001). "PEP 7 – Style Guide for C Code". Python Enhancement Proposals. Python Software Foundation. Retrieved 24 November 2008

- [2] Piotrowski, Przemyslaw (July 2006). "Build a Rapid Web Development Environment for Python Server Pages and Oracle". Oracle Technology Network. Oracle. Retrieved 12 March 2012.
- [3] Blogger "The History of Python: A Brief Timeline of Python". 20 January 2009. Retrieved 20 March 2016.
- [4] Deily, Ned (19 December 2017). "Python 3.6.4 is now available". Python Insider. The Python Core Developers. Retrieved 20 December 2017.
- [5] Roy Greenfeld, Daniel; Roy Greenfeld, Audrey (2015), Two Scoops of Django: Best Practices for Django 1.8 (3rd ed.).
- [6] Allen, Grant; Owens, Mike (November 5, 2010). The Definitive Guide to SQLite (2nd ed.).

**WEBSITES+**

- [1] Tutorials point, <http://www.tutorialspoint.com/asp.net/>
- [2] W3schools, <http://www.w3schools.in/python-tutorial/>
- [3] Guru99, <http://www.guru99.com/python-tutorials.html>
- [4] Django for beginners, <http://www.djangoforbeginners.com/>
- [5] SQLite tutorial ,<http://www.sqlitetutorial.net>

**APPENDIX**

**A.SCREENSHOT**

**RUNNING PROCESS**



FIG-4.a)  
**REGISTRATION PAGE**



FIG-4.b)

REGISTERED REPORT

Firstname	Lastname	Citizen_name	Aadhar_id	Mobile_number	Age	Gender	Email_id	Address	DOB	Action
boopathi	kumar	kumar boopathi	845267415874	9846578421	45	male	boopathi@gmail.com	coimbatore	20.06.1972	<a href="#">Edit</a> <a href="#">Delete</a>
mohan	balaji	balaji mohan	845698754684	9875469875	22	male	balaji@gmail.com	setam	16.02.1995	<a href="#">Edit</a> <a href="#">Delete</a>
rohan	balaji	rohan balaji	7658451257	9857884565	24	male	rohan@gmail.com	erode	02.06.1993	<a href="#">Edit</a> <a href="#">Delete</a>
seema	raj	seema raj	754268451454	9874562214	23	female	seema@gmail.com	colmatore	05.01.1994	<a href="#">Edit</a> <a href="#">Delete</a>
Guru	raj	guru raj	786548798822	9854655254	22	male	guru53@gmail.com	erode	06.07.1995	<a href="#">Edit</a> <a href="#">Delete</a>
rahul	raj	rahul raj	845745654125	9845745874	22	male	rahul654@gmail.com	erode	10.12.1995	<a href="#">Edit</a> <a href="#">Delete</a>

FIG-4.c) RUNNING PROCESS

```
Microsoft Windows [Version 6.0.6002.18005] Copyright (c) 2018 Microsoft Corporation. All rights reserved.  
C:\Users\DELL>  
E:\>cd project\saving form\django project-claimweb  
E:\project\saving form\django project-claimweb>  
E:\project\saving form\django project-claimweb>dir  
Volume in drive E: has no label.  
Volume Serial number is J4D8-568D  
  
Directory of E:\project\saving form\django project-claimweb  
03-03-2018  04:44 PM  <DIR>      .  
03-03-2018  04:44 PM  <DIR>      ..  
03-03-2018  04:32 PM  <DIR>      crud  
03-03-2018  04:44 PM                139,264 db.sqlite3  
24-06-2017  07:03 AM                813 manage.py  
03-03-2018  04:32 PM  <DIR>      web  
                2 File(s)      148,887 bytes  
                4 Dir(s)  35,926,225,944 bytes free  
  
E:\project\saving form\django project-claimweb>python manage.py runserver  
Performing system checks...  
  
System check identified some issues:  
  
WARNINGS:  
?: (1/1000) MIDDLEWARE_CLASSES is not set.  
-----  
NOTE: Django 1.7 changed the global defaults for the MIDDLEWARE_CLASSES. django.contrib.sessions.middleware.SessionMiddleware, django.contrib.auth.middleware.AuthenticationMiddleware, and django.contrib.messages.middleware.MessageMiddleware were removed from the defaults. If your project needs these middleware then you should configure this setting.  
  
System check identified 1 issue (0 silenced).  
March 28, 2018 - 06:35:03  
Django version 1.8.1, using settings 'web.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

FIG-5.a)



CLAIM PAGE



FIG-5.b)  
SAVINGS REPORT

Firstname	Lastname	Citizen_name	Aadhar_id	Salary	Annual_income	Address	Current_savings	Status	Action
boopathi	kumar	boopathi kumar	845745654125	300000	550000	erode	1500000	Available	<a href="#">Edit</a> <a href="#">Delete</a>
rohan	balaji	rohan balaji	823146512346	25000	500000	ram nagar coimbatore	1000000	Available	<a href="#">Edit</a> <a href="#">Delete</a>
mohan	balaji	mohan balaji	854123651243	25000	500000	balaji nagar,coimbatore	600000	Not Available	<a href="#">Edit</a> <a href="#">Delete</a>
rahul	raj	rahul raj	821345124512	45000	200000	15000	500000	Available	<a href="#">Edit</a> <a href="#">Delete</a>

FIG-5.c)

**RUNNING PROCESS**

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\DELL>

E:\>cd project\saving form\death\usb

E:\project\saving form\death\web>dir
Volume in drive E has no label.
Volume Serial Number is 2400-5080

Directory of E:\project\saving form\death\web

18-03-2018  06:05 AM    <DIR>          .
18-03-2018  06:05 AM    <DIR>          ..
18-03-2018  05:58 AM    <DIR>          crud
18-03-2018  06:05 AM    130,264  db.sqlite3
24-08-2017  07:03 AM    <DIR>          manage.py
18-03-2018  10:22 PM    <DIR>          web
                2 File(s)      140,087 bytes
                4 Dir(s)   15,826,456,320 bytes free

E:\project\saving form\death\web>path c:\python27

E:\project\saving form\death\web>python manage.py runserver
Performing system checks...

System check identified some issues:

WARNINGS:
?: (1.7.WARN) MIDDLEWARE_CLASSES is not set.
   HINT: Django 1.7 changed the global defaults for the MIDDLEWARE_CLASSES. django.contrib.sessions.middleware.SessionMiddleware, django.contrib.auth.middleware.AuthenticationMiddleware, and django.contrib.messages.middleware.MessageMiddleware were removed from the defaults. If your project needs these middleware then you should configure this setting.

System check identified 1 issue (0 silenced).
March 20, 2018 - 06:10:50
Django version 1.8.1, using settings 'web.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

FIG-6.a)

**DEATH UPDATE**



FIG-6.b)

CERTIFICATE REPORT

Date of death	Gender	Name	Father/ Husband	Place of death	Aadhar_id	Action
2018-03-02	male	mohan	chinnaiya	selam	845236541452	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2018-01-25	female	Seema	pandfyan	coimbatore	854123651243	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2017-12-18	male	yuvaraj	nagaraj	theni	823146512346	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2017-10-20	male	naveen	salvam	Erode	821345124512	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

FIG-6.c)

UPDATE PROCESS

127.0.0.1:8000/crud/edit/12

127.0.0.1:8000/crud/edit/12

Firstname  
rahul

Lastname  
raj

Citizen\_name  
rahul raj

Aadhar\_id  
84574554125

Mobile\_number  
9845745874

Age  
22

Gender  
male

Email\_id  
rahun854@gmail.com

Address  
erode

DOB  
10.12.1995

FIG-7.a)

127.0.0.1:8000/crud/edit/2

Firstname  
boopathi

Lastname  
kumar

Citizen\_name  
boopathi kumar

Aadhar\_id  
845745654125

Salary  
300000

Annual\_income  
550000

Address  
erode

Current\_savings  
1500000

Status  
Available

Update

FIG-7.b)

127.0.0.1:8000/crud/edit/14

## Death certificate update

Date of death  
2017-10-20

Gender  
male

Name  
sivan

Father/Husband name  
sivan

Place of death  
Erode

Aadhar\_id  
845745654125

Update

FIG-7.c)

## B.SAMPLE CODE

### Models.py

From django.db import models

# Create your models here.

Class Member (models. Model):

    firstname = models.CharField(max\_length=40)

    lastname = models.CharField(max\_length=40)

    citizen\_name = models.CharField(max\_length=40)

    aadhar\_id = models.CharField(max\_length=40)

    mobile\_number = models.CharField(max\_length=40)

    age = models.CharField(max\_length=40)

```

gender = models.CharField(max_length=40)
email_id = models.CharField(max_length=40)
address = models.CharField(max_length=40)
dob = models.CharField(max_length=40)
def __str__(self):
    return self.firstname + " " + self.lastname + " " + self.citizen_name + " " + self.aadhar_id + " " + self.mobile_number + " " + self.age + " " +
self.gender + " " + self.email_id + " " + self.address + " " + self.dob

```

**urls.py**

```

from django.conf.urls import url
from . import views
urlpatterns= [
    url(r'^$', views.index, name='index'),
    url(r'^create$', views.create, name='create'),
    url(r'^edit/(?P<id>\d+)$', views.edit, name='edit'),
    url(r'^edit/update/(?P<id>\d+)$', views.update, name='update'),
    url(r'^delete/(?P<id>\d+)$', views.delete, name='delete'),
]

```

**Views.py**

```

from django.shortcuts import render, redirect
from .models import Member
# Create your views here.
def index(request):
    members = Member.objects.all()
    context = {'members': members}
    return render(request, 'crud/index.html', context)
def create(request):
    member = Member(firstname=request.POST['firstname'], lastname=request.POST['lastname'], citizen_name=request.POST['citizen_name'],
aadhar_id = request.POST['aadhar_id'], mobile_number=request.POST['mobile_number'], age=request.POST['age'],
gender=request.POST['gender'], email_id=request.POST['email_id'], address=request.POST['address'], dob=request.POST['dob'])
    member.save()
    return redirect('/')
def edit(request, id):
    members = Member.objects.get(id=id)
    context = {'members': members}
    return render(request, 'crud/edit.html', context)
def update(request, id):
    member = Member.objects.get(id=id)
    member.firstname = request.POST['firstname']
    member.lastname = request.POST['lastname']
    member.citizen_name = request.POST['citizen_name']
    member.aadhar_id = request.POST['aadhar_id']
    member.mobile_number = request.POST['mobile_number']
    member.age = request.POST['age']
    member.gender = request.POST['gender']
    member.email_id = request.POST['email_id']
    member.address = request.POST['address']
    member.dob = request.POST['dob']
    member.save()
    return redirect('/crud/')
def delete(request, id):
    member = Member.objects.get(id=id)
    member.delete()
    return redirect('/crud/')

```

**Base.html**

```

<!DOCTYPE html>
<html lang="en">
<head>

```



```
<meta charset="UTF-8" name="viewport" content="width=device-width, initial-scale=1"/>
```

```
{% load static %}
```

```
<link rel="stylesheet" type="text/css" href="{% static 'crud/css/bootstrap.css' %}"/>
```

```
<style>
```

```
table
```

```
{
border-collapse: collapse;
width: 100%;
border-color: black;
}
```

```
th, td
```

```
{
text-align: left;
padding: 8px;
border-color: black;
color: white;
}
```

```
tr:nth-child(even){background-color: #2C3E50}
```

```
tr:nth-child(odd){background-color: #1C2833}
```

```
th
```

```
{
background-color: #1D8348;
color: white;
border-color: black;
}
```

```
input[type=text]
```

```
{
width: 100%;
padding: 12px 20px;
margin: 8px 0;
box-sizing: border-box;
border: none;
background-color: #D5DBDB;
color: #2874A6;
}
```

```
label
```

```
{
color: #E5E8E8;
}
```

```
body {
```

```
background:
```

```
linear-gradient(27deg, #151515 5px, transparent 5px) 0 5px,
```

```
linear-gradient(207deg, #151515 5px, transparent 5px) 10px 0px,
```

```
linear-gradient(27deg, #222 5px, transparent 5px) 0px 10px,
```

```
linear-gradient(207deg, #222 5px, transparent 5px) 10px 5px,
```

```
linear-gradient(90deg, #1b1b1b 10px, transparent 10px),
```

```
linear-gradient(#1d1d1d 25%, #1a1a1a 25%, #1a1a1a 50%, transparent 50%, transparent 75%, #242424 75%, #242424);
```

```
background-color: #131313;
```

```
background-size: 20px 20px;
```

```
}
```

```
.gradient-pattern {
```

```
-webkit-box-sizing: content-box;
```

```
-moz-box-sizing: content-box;
```

```
box-sizing: content-box;
```

```
width: 1350px;
```

```
height: 75px;
```

```
border: none;
```

```
font: normal 0.5px/normal Arial, Helvetica, sans-serif;
```

```
color: rgba(255,255,255,1);
```

```
-o-text-overflow: clip;
```



```

text-overflow: clip;
background: -webkit-linear-gradient(135deg, rgba(0,0,0,0) 25%, rgba(255,255,255,0.2) 25%, rgba(255,255,255,0.2) 50%, rgba(0,0,0,0) 50%,
rgba(0,0,0,0) 75%, rgba(255,255,255,0.2) 75%), -webkit-linear-gradient(45deg, rgba(0,0,0,0) 25%, rgba(255,255,255,0.2) 25%,
rgba(255,255,255,0.2) 50%, rgba(0,0,0,0) 50%, rgba(0,0,0,0) 75%, rgba(255,255,255,0.2) 75%), rgb(0, 153, 151);
background: -moz-linear-gradient(-45deg, rgba(0,0,0,0) 25%, rgba(255,255,255,0.2) 25%, rgba(255,255,255,0.2) 50%, rgba(0,0,0,0) 50%,
rgba(0,0,0,0) 75%, rgba(255,255,255,0.2) 75%), -moz-linear-gradient(45deg, rgba(0,0,0,0) 25%, rgba(255,255,255,0.2) 25%,
rgba(255,255,255,0.2) 50%, rgba(0,0,0,0) 50%, rgba(0,0,0,0) 75%, rgba(255,255,255,0.2) 75%), rgb(0, 153, 151);
background: linear-gradient(-45deg, rgba(0,0,0,0) 25%, rgba(255,255,255,0.2) 25%, rgba(255,255,255,0.2) 50%, rgba(0,0,0,0) 50%,
rgba(0,0,0,0) 75%, rgba(255,255,255,0.2) 75%), linear-gradient(45deg, rgba(0,0,0,0) 25%, rgba(255,255,255,0.2) 25%, rgba(255,255,255,0.2)
50%, rgba(0,0,0,0) 50%, rgba(0,0,0,0) 75%, rgba(255,255,255,0.2) 75%), rgb(0, 153, 151);
background-position: auto auto;
-webkit-background-origin: padding-box;
background-origin: padding-box;
-webkit-background-clip: border-box;
background-clip: border-box;
-webkit-background-size: 200em 200em;
background-size: 200em 200em;
}
.inset-text-effect {
display: inline-block;
-webkit-box-sizing: content-box;
-moz-box-sizing: content-box;
box-sizing: content-box;
border: none;
font: normal 35px/1 "Aladin", Helvetica, sans-serif;
color: rgba(48,48,48,1);
text-align: center;
text-transform: normal;
-o-text-overflow: clip;
text-overflow: clip;
white-space: pre;
text-shadow: 1px 1px 0 rgba(140,140,140,0.6) , -1px -1px 1px rgba(0,0,0,0.67) ;
}
.inset-text-effects {
display: inline-block;
-webkit-box-sizing: content-box;
-moz-box-sizing: content-box;
box-sizing: content-box;
border: none;
font: normal 50px/1 "Aladin", Helvetica, sans-serif;
color: rgba(48,48,48,1);
text-transform: normal;
-o-text-overflow: clip;
text-overflow: clip;
white-space: pre;
text-shadow: 1px 1px 0 rgba(140,140,140,0.6) , -1px -1px 1px rgba(0,0,0,0.67) ;
}
</style>
</head>
<body>
<nav class="navbar navbar-default" style="background-color:#AEB6BF;">
<div class="gradient-pattern">
<a class="navbar-brand" href=""><font face="Comic Sans MS" color= "#212F3D"><div class="inset-text-effect"> Project on Web
Based Personal Savings Claim Process</div></font></a>
</div>
</nav>
<div class="col-md-1"></div>
<div class="col-md-10 well" style="background-color:#465654;">
<h3 class="text-primary">
<div class="inset-text-effects">
Citizen Registration </div> </h3>
<hr style="border-top:1px dotted #000;">

```







```

        <th>Gender</th>
        <th>Email_id</th>
        <th>Address</th>
        <th>DOB</th>
    <th>Action</th>
</tr>
</thead>
<tbody>
    {% for member in members %}
    <tr>
        <td>{{ member.firstname }}</td>
        <td>{{ member.lastname }}</td>
        <td>{{ member.citizen_name }}</td>
        <td>{{ member.aadhar_id }}</td>
        <td>{{ member.mobile_number }}</td>
        <td>{{ member.age }}</td>
        <td>{{ member.gender }}</td>
        <td>{{ member.email_id }}</td>
        <td>{{ member.address }}</td>
        <td>{{ member.dob }}</td>
        <td><center><a class="btn btn-sm btn-warning" href="edit/{{ member.id }}"><span class="glyphicon glyphicon-edit"></span>
Edit</a> <a class="btn btn-sm btn-danger" href="delete/{{ member.id }}"><span class="glyphicon glyphicon-trash"></span>
Delete</a></center></td>
    </tr>
    {% endfor %}
</tbody>
</table>
{% endblock %}

```

### Edit.html

```

{% extends 'crud/base.html' %}
{% block body %}
<form method="POST" action="update/{{ members.id }}">
    {% csrf_token %}
    <div class="form-group">
        <label for="firstname">Firstname</label>
        <input type="text" name="firstname" value="{{ members.firstname }}" required="required"/>
    </div>
    <div class="form-group">
        <label for="lastname">Lastname</label>
        <input type="text" name="lastname" value="{{ members.lastname }}" required="required"/>
    </div>
    <div class="form-group">
        <label for="citizen_name">Citizen_name</label>
        <input type="text" name="citizen_name" value="{{ members.citizen_name }}" required="required"/>
    </div>
    <div class="form-group">
        <label for="aadhar_id">Aadhar_id</label>
        <input type="text" name="aadhar_id" value="{{ members.aadhar_id }}" required="required"/>
    </div>
    <div class="form-group">
        <label for="mobile_number">Mobile_number</label>
        <input type="text" name="mobile_number" value="{{ members.mobile_number }}" required="required"/>
    </div>
    <div class="form-group">
        <label for="age">Age</label>
        <input type="text" name="age" value="{{ members.age }}" required="required"/>
    </div>
    <div class="form-group">
        <label for="gender">Gender </label>
        <input type="text" name="gender" value="{{ members.gender }}" required="required"/>
    </div>

```

```
</div>
    <div class="form-group">
    <label for="email_id">Email_id</label>
    <input type="text" name="email_id" value="{{ members.email_id }}" required="required"/>
</div>
    <div class="form-group">
    <label for="address">Address </label>
    <input type="text" name="address" value="{{ members.address }}" required="required"/>
</div>
    <div class="form-group">
    <label for="dob">DOB </label>
    <input type="text" name="dob" value="{{ members.dob }}" required="required"/>
</div>

<div class="form-group">
    <button type="submit" class="btn btn-sm btn-warning"><span class="glyphicon glyphicon-edit"></span> Update</button>
</div>
</form>
{% endblock %}
```

