

Rural Water Supply and Sanitation System



BTech/III Year CSE/V Semester

15CSE302/Database Management Systems

Project Review -3

Team Syntax_Error

Rollno	Name
CB.EN.U4CSE18435	Maddu Jaya Sai Durga Akhil
CB.EN.U4CSE18437	Mandru Vinay
CB.EN.U4CSE18440	C Monishver
CB.EN.U4CSE18443	Potti Shanmuka Abhinay

Amrita School of Engineering, Coimbatore
Department of Computer Science and Engineering
2020 -2021 Odd Semester

Table of Contents

Title	Page number
Chapter 1 Abstract	4
Chapter 2 Business rules	5
Tables	
Output	
Chapter 3 Preview for the project	7
Introduction	
Need and motivation	
Tools used	
Chapter 4 Project analysis	8
List of modules in the project	
Module-wise explanation	
Chapter 5 Project design	10
ER-Diagram	
Database scheme before normalization	
Chapter 6 Normalisation	11
Normalization procedure	
Database schema after normalization	
Chapter 7 Backend design	49
Table creation comments	
Sample instances of the tables	
Chapter 8 Frontend design	62
Introduction to the tool(s)(used in the project)	
Connectivity code	

Chapter 9 Database connectivity	70
Introduction to the connectivity standard (used in the project)	
Connectivity code	
Chapter 10 Sample code	74
Sample code of selected UI screens	
Chapter 11 Conclusions	81
Elucidating important features of the project	
Chapter 12 References	81

Chapter 1 Abstract

Project Title : Rural Water Supply and Sanitation

Abstract:

The Rural Water Supply and Sanitation Project seek to improve access by rural populations to safe potable water and to support implementation of a program to supply water to rural populations. There are four components. For the first component, new and rehabilitated and improved water supply facilities will be installed in communities, villages, and groups of villages of stated populations. The community participation approach will integrate the provision of safe drinking water, sanitation services, and health and hygiene education. The second component consists of making sanitation investments; constructing safe wastewater and excreta disposal systems; using appropriate technology; a hygiene education program; and providing public installations on a pilot basis, such as block latrines for schools, small sewer collectors, and community septic tanks, as well as lagoon-type wastewater treatment. The participatory approach, component three, will include community mobilization and health education, carried out mainly by local personnel. The fourth component will help provide technical back-stopping, monitoring, and evaluation for the decentralized project implementation by enlisting the services of local and international consultants.

Chapter 2 Business rules

Business Rules:

- We have two different types of users :a)official b)public
- public can view and request for new projects
- officials can edit and validate their request.
- sanitation services like facility of safe drinking water,health and hygiene education in districts,mandals and panchayats.
- local personnels can approve or reject implementation on sanitation facilities.
- Non-profit organisations can fund for the projects
- continuous monitoring on all projects

Tables:

- District
- Mandal
- Panchayat
- Organization
- Project_finacial details
- Excreta disposal system details
- Registrations
- User
- Employee
- Health and hygiene education details
- Individual house hold latrine details
- Safe water treatment details
- Work status defined by roles

Output:

- A web application to register new sanitation project, to accept fund for the projects, to monitor the process of approval and financial details of Rural water supply and sanitation system, to monitor the technical background of those areas sorted by local and international consultants.

Chapter 3 Preview for the project

Introduction:

The main objective of Rural Water Supply and Sanitation Project seek to improve access by rural populations to safe potable water and to support implementation of a program to supply water to rural populations The Rural Water Supply and Sanitation Project seek to improve access by rural populations to safe potable water and to support implementation of a program to supply water to rural populations

Need and motivation:

The Rural Water Supply and Sanitation Project seek to improve access by rural populations to safe potable water and to support implementation of a program to supply water to rural populations.

Tools used:

- HTML5
- CSS3
- JavaScript
- Bootstrap
- Django
- JQuery
- Postgresql

Chapter 4 Project analysis

List of modules in the project:

New project registration

Monitor financial details

User login registration

Rural water supply and sanitation registrations

Entity - Set Model:

Entities

1)VillageDetails

2)Water_Source_details

3)Quality_Assurance

4)Filtration

5)Water_Revenue_Details

6)Pipeline_Details

7)Water_Connections

8)Individual_House_Hold_Latrines

9)IHHL_Expenditure_Details

10)Sanction

11)Deposit_Works

12)Requirements

13)Users

14)Administrator

15)EmployeeDetails

Attributes

Administrator	Deposit_Works	EmployeeDetails
* <u>admin_id</u> °admin_password °status	* <u>works_id</u> °SanctionID °Max transportation of water	* <u>emp_id</u> °salary °type °age

Filtration	Users	Water_Source_Details
* <u>filtration_id</u> °Filtration_Method °Location °Capacity °Maintenance review date	* <u>userID</u> °Password °Fname °Lname °location_address °VillageID	* <u>source_id</u> °Source_Name °Source_Capacity °Maintenance Review Date

Individual_House_Hold_Latrines	IHHL_Expenditure_Details	Pipeline_Details
* <u>IHHL_id</u> °VillageID °sanctioned_IHHL °inprogress_IHHL °Completed IHHL	* <u>IHHL_fund_id</u> °sanctioned_IHHLs °Estimated_Cost °funds from ngo	°Length_of_pipeline °Year_of_laying °Pipe type

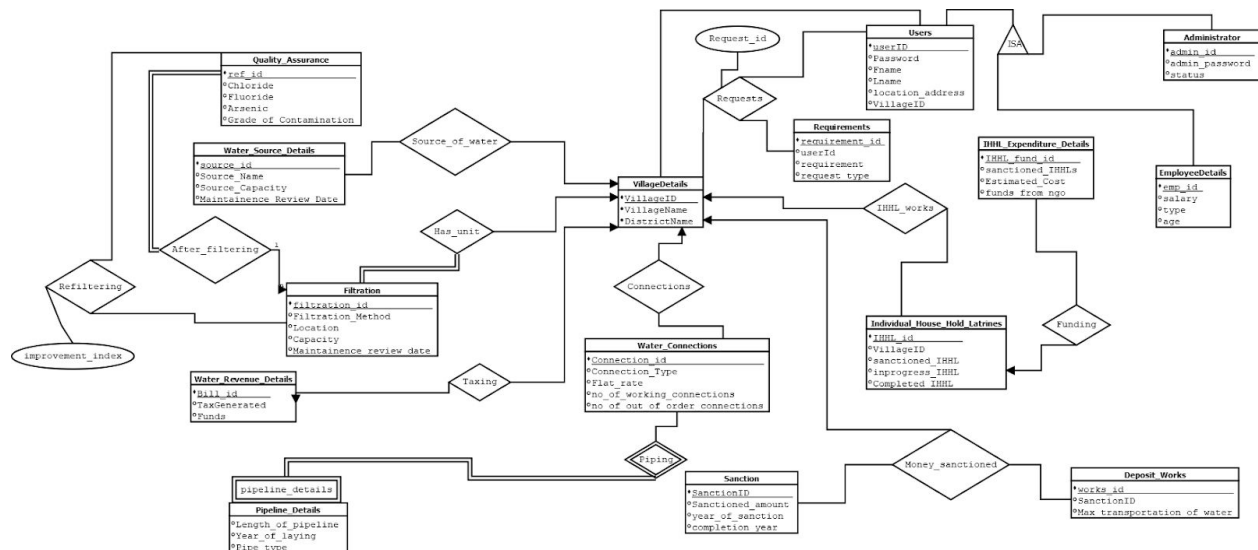
Quality_Assurance	Quality_Assurance
°Chloride °Fluoride °Arsenic °Grade of Contamination	* <u>ref_id</u> °Chloride °Fluoride °Arsenic °Grade of Contamination

Requirements	Sanction	VillageDetails
* <u>requirement_id</u> °userId °requirement °request type	* <u>SanctionID</u> °Sanctioned_amount °year_of_sanction °completion year	* <u>VillageID</u> °VillageName °DistrictName

Water_Connections	Water Revenue Details
* <u>Connection_id</u> °Connection_Type °Flat_rate °no_of_working_connections °no of out of order connections	* <u>Bill_id</u> °TaxGenerated °Funds

Chapter 5 Project design

ER Diagram:



Database Scheme before Normalization:

1. Write out the relation including all attribute names. Write down all Functional Dependencies. and draw dependency diagram

Master Table:

(village_id, village_name, district_name, Source_id, Source_name, Source_capacity, Maintenance_review_dates, Village_id, filtration_id, Filtration_method, filtration_location, filtration_capacity, Maintainance_review_date, ref_id, chloride, fluoride, arsenic, grade_of_contamination, ref_id, filtration_id, improvement_index, Bill_id, Tax_generated, funds, village_id, Connection_id, Connection_type, Flat_rate, no_of_working_connections, no_of_out_of_connections, Village_id, connection_id, length_of_pipeline, year_of_laying, pipe_type, sanction_id, deposit_works, village_id, Sanction_id, sanctioned_amount, year_of_sanction, completion_year, Works_id, sanction_id, max_transportation_of_water, ihhl_id, village_id, sanctioned_ihhls, inprogress_IHHL, Completed_IHHL, sanction_amt, IHHL_fund_id, Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, IHHL_id, Request_id, requirement_id, user_id, village_id, requirement_id, user_id, requirement, request_type, User_id, user_pwd, Fname, Lname, Location_address, village_id, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age)

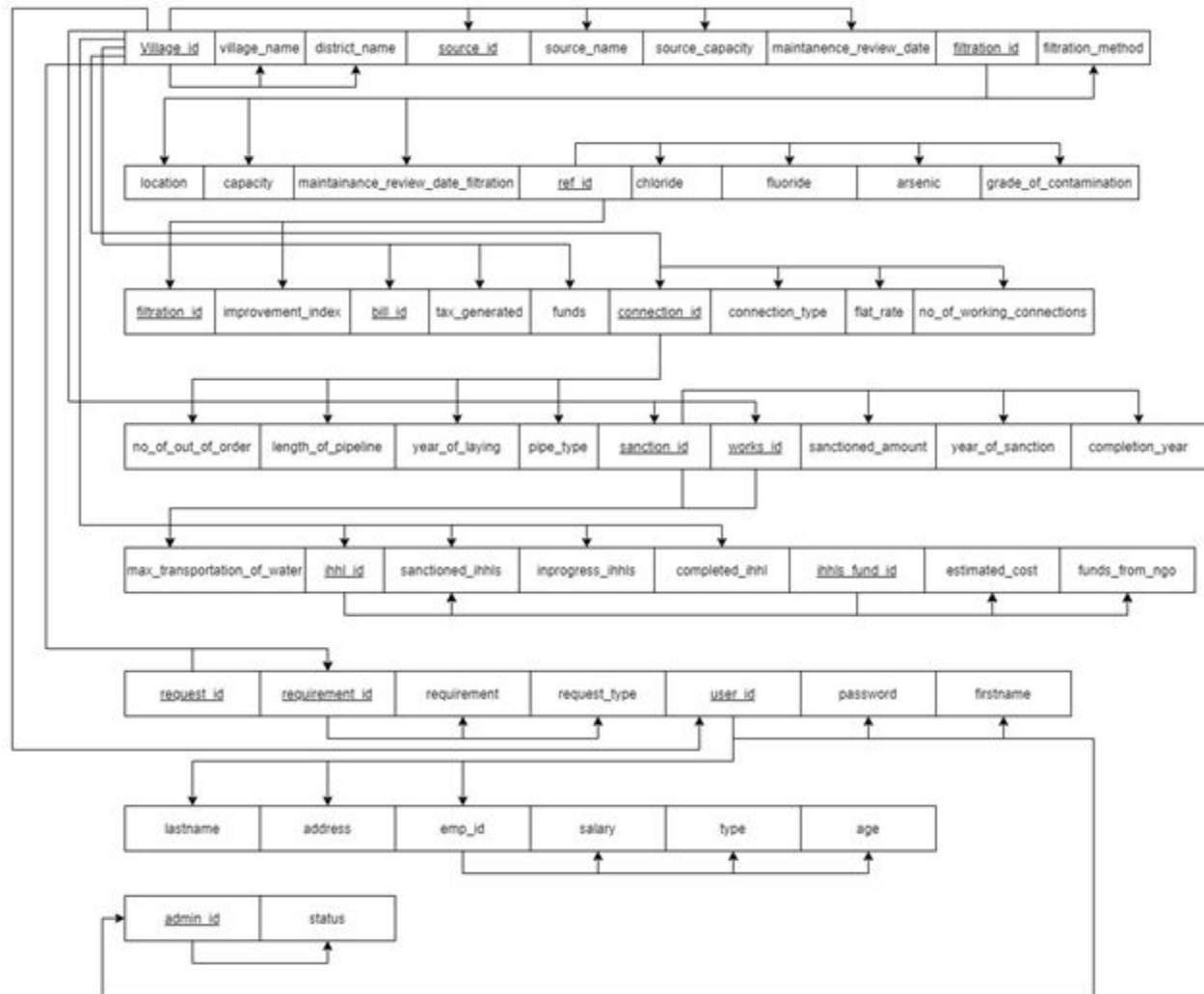
Chapter 6 Normalization

Normalization procedure:

Identifying all the Functional Dependencies :

1. (Source_id) → (Source_name)
2. (Source_id) → (Source_capacity)
3. (Source_id) → (Maintenance_review_start_date)
4. (Source_id) → (Maintenance_review_end_date)
5. (filtration_id) → (Filtration_method)
6. (Connection_id, Connection_type) → (flat_rate)
7. (IHHL_id, IHHL_fund_id) → (village_id)
8. (Request_id, requirement_id, User_id, Admin_id, Emp_id) → (requirement)
9. (Admin_id, Request_id, requirement_id) → (project_status)
10. (village_id) → (village_name, district_name)
11. (Source_id) → (Source_name, Source_capacity,
Maintenance_review_start_date, Maintenance_review_end_date)
12. (filtration_id) → (filtration_location, filtration_capacity)
13. (ref_id) → (chloride, fluoride, arsenic, grade_of_contamination)
14. (Bill_id) → (Tax_generated, funds)
15. (Connection_id) → (no_of_working_connections,
no_of_out_of_connections)

- 16.(Connection_id) → (length_of_pipeline, year_of_laying, pipe_type)
- 17.(Sanction_id) → (Sanctioned_amount, year_of_sanction, completion_year)
- 18.(Sanction_id) → (deposit_works)
- 19.(Works_id) → (max_transportation_of_water)
- 20.(IHHL_id) → (inprogress_IHHL, Completed_IHHL, sanction_amt)
- 21.(IHHL_fund_id) → (Sanctioned_IHHLs, Estimated_cost, funds_from_ngo)
- 22.(Request_id, requirement_id) → (request_type)
- 23.(Request_id, requirement_id) → (village_id)
- 24.(User_id) → (user_pwd, Fname, Lname, Location_address, village_id)
- 25.(Admin_id) → (admin_pwd)
- 26.(Emp_id) → (emp_pwd, emp_type, emp_age)
- 27.(emp_type) → (emp_salary)



2) Finding the FD closure and Attribute Closure

a) FD closure:

1. Rural_water(village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, Works_id)

FD's : No functional dependencies, as this table is used for referring and connecting other tables.

FD Closure : None.

Attribute Closure : For all attributes, since no functional dependencies, the

attribute closure for each is same as the attribute itself.

2. Village_Details (village_id, village_name, district_name)

FD's : (village_id) \rightarrow (village_name, district_name)

FD Closure : (village_id) \rightarrow (village_name), (village_id) \rightarrow (district_name).

Attribute Closure :

1. (village_id)+ = (village_id, village_name, district_name)

3. Water_source_details (Source_id, Source_name, Source_capacity,

Maintenance_review_start_date, Maintenance_review_end_date)

FD's : (Source_id) \rightarrow (Source_name), (Source_id) \rightarrow (Source_capacity),

(Source_id) \rightarrow (Maintenance_review_start_date), (Source_id) \rightarrow

(Maintenance_review_end_date)

FD Closure : (Source_id) \rightarrow (Source_name, Source_capacity,

Maintenance_review_start_date, Maintenance_review_end_date),

(Source_id) \rightarrow (Source_name, Source_capacity), (Source_id) \rightarrow

(Maintenance_review_start_date, Maintenance_review_end_date)

Attribute Closure :

• (Source_id)+ = (Source_id, Source_name, Source_capacity,

Maintenance_review_start_date, Maintenance_review_end_date)

4. Filtration_method (filtration_id, Filtration_method)

FD's : (filtration_id) \rightarrow (Filtration_method)

FD Closure : (filtration_id) → (Filtration_method)

Attribute Closure :

- (filtration_id)+ = (filtration_id, Filtration_method)

5. Filtration_details (filtration_id, filtration_location, filtration_capacity)

FD's : (filtration_id) → (filtration_location, filtration_capacity)

FD Closure : (filtration_id) → (filtration_location), (filtration_id) →
(filtration_capacity)

Attribute Closure :

- (filtration_id)+ = (filtration_id, filtration_location, filtration_capacity)

6. Quality_assurance (ref_id, chloride, fluoride, arsenic,
grade_of_contamination)

FD's : (ref_id) → (chloride, fluoride, arsenic, grade_of_contamination)

FD Closure : (ref_id) → (chloride, fluoride, arsenic), (ref_id) →
(grade_of_contamination)

Attribute Closure :

- (ref_id)+ = (ref_id, chloride, fluoride, arsenic,
grade_of_contamination)

7. Water_revenue_details (Bill_id, Tax_generated, funds)

FD's : (Bill_id) → (Tax_generated, funds)

FD Closure : (Bill_id) → (Tax_generated), (Bill_id) → (funds)

Attribute Closure :

- $(\text{Bill_id})^+ = (\text{Bill_id}, \text{Tax_generated}, \text{funds})$

8. Village_water_connections(Connection_id, no_of_working_connections,
no_of_out_of_connections)

FD's : $(\text{Connection_id}) \rightarrow (\text{no_of_working_connections},$
 $\text{no_of_out_of_connections})$

FD Closure : $(\text{Connection_id}) \rightarrow (\text{no_of_working_connections},$
 $(\text{Connection_id}) \rightarrow (\text{no_of_out_of_connections})$

Attribute Closure :

- $(\text{Connection_id})^+ = (\text{Connection_id}, \text{no_of_working_connections},$
 $\text{no_of_out_of_connections})$

9. Water_Connection_type (Connection_id, Connection_type, Flat_rate)

FD's : $(\text{Connection_id}, \text{Connection_type}) \rightarrow (\text{flat_rate})$

FD Closure : $(\text{Connection_id}, \text{Connection_type}) \rightarrow (\text{flat_rate})$

Attribute Closure :

- $(\text{Connection_id})^+ = (\text{Connection_id}, \text{flat_rate})$
- $(\text{Connection_type})^+ = (\text{Connection_type}, \text{flat_rate})$

10. Pipeline_details (Connection_id, length_of_pipeline, year_of_laying,
pipe_type)

FD's : $(\text{Connection_id}) \rightarrow (\text{length_of_pipeline}, \text{year_of_laying}, \text{pipe_type})$

FD Closure : $(\text{Connection_id}) \rightarrow (\text{length_of_pipeline}), (\text{Connection_id}) \rightarrow$

$(\text{year_of_laying}), (\text{Connection_id}) \rightarrow (\text{pipe_type})$

Attribute Closure :

- $(\text{Connection_id}) \rightarrow (\text{Connection_id}, \text{length_of_pipeline}, \text{year_of_laying}, \text{pipe_type})$

11.Sanction ($\text{Sanction_id}, \text{sanctioned_amount}, \text{year_of_sanction}, \text{completion_year}$)

FD's : $(\text{Sanction_id}) \rightarrow (\text{Sanctioned_amount}, \text{year_of_sanction}, \text{completion_year})$

FD Closure : $(\text{Sanction_id}) \rightarrow (\text{Sanctioned_amount}), (\text{Sanction_id}) \rightarrow (\text{year_of_sanction}), (\text{Sanction_id}) \rightarrow (\text{completion_year})$

Attribute Closure :

- $(\text{Sanction_id}) \rightarrow (\text{Sanction_id}, \text{Sanctioned_amount}, \text{year_of_sanction}, \text{completion_year})$

12.works ($\text{works_id}, \text{max_transportation_of_water}$)

FD's : $(\text{Works_id}) \rightarrow (\text{max_transportation_of_water})$

FD Closure : $(\text{Works_id}) \rightarrow (\text{max_transportation_of_water})$

Attribute Closure :

- $(\text{Works_id}) \rightarrow (\text{Works_id}, \text{max_transportation_of_water})$

13.Depositworks ($\text{Sanction_id}, \text{deposit_works}$)

FD's : (Sanction_id) \rightarrow (deposit_works)

FD Closure : (Sanction_id) \rightarrow (deposit_works)

Attribute Closure :

- (Sanction_id) \rightarrow (Sanction_id, deposit_works)

14.village_IHHL (IHHL_id, IHHL_fund_id, village_id)

FD's : (IHHL_id, IHHL_fund_id) \rightarrow (village_id)

FD Closure : (IHHL_id) \rightarrow (village_id),

(IHHL_fund_id) \rightarrow (village_id)

Attribute Closure :

- (IHHL_id) \rightarrow (IHHL_id, village_id)
- (IHHL_fund_id) \rightarrow (IHHL_fund_id, village_id)

15.IHHL_expenditure_details (IHHL_fund_id, Sanctioned_IHHLs,

Estimated_cost, funds_from_ngo)

FD's : (IHHL_fund_id) \rightarrow (Sanctioned_IHHLs, Estimated_cost,

funds_from_ngo)

FD Closure :(IHHL_fund_id) \rightarrow (Sanctioned_IHHLs), (IHHL_fund_id) \rightarrow

(Estimated_cost), (IHHL_fund_id) \rightarrow (funds_from_ngo)

Attribute Closure :

- (IHHL_fund_id)+ = (IHHL_fund_id, Sanctioned_IHHLs,

Estimated_cost, funds_from_ngo)

16.Individual_house_hold_latrines (IHHL_id, inprogress_IHHL,

Completed_IHHL, sanction_amt)

FD's : (IHHL_id) \rightarrow (inprogress_IHHL, Completed_IHHL, sanction_amt)

FD Closure : (IHHL_id) \rightarrow (inprogress_IHHL), (IHHL_id) \rightarrow

(Completed_IHHL), (IHHL_id) \rightarrow (sanction_amt)

Attribute Closure :

- (IHHL_id)+=(IHHL_id, inprogress_IHHL, Completed_IHHL, sanction_amt)

17.village_request(Request_id, requirement_id, village_id)

FD's : (Request_id, requirement_id) \rightarrow (village_id)

FD Closure : (Request_id) \rightarrow (village_id), (requirement_id) \rightarrow (village_id)

Attribute Closure :

- (Request_id)+=(Request_id, village_id)
- (requirement_id)+=(requirement_id, village_id)

18.Requirements (Request_id, requirement_id, request_type)

FD's : (Request_id, requirement_id) \rightarrow (request_type)

FD Closure : (Request_id) \rightarrow (request_type), (requirement_id) \rightarrow

(request_type)

Attribute Closure :

- (Request_id) \rightarrow (Request_id, request_type)

- (requirement_id) → (requirement_id, request_type)

19. User_requests (Request_id, requirement_id, User_id, Admin_id, Emp_id, requirement)

FD's : (Request_id, requirement_id, User_id, Admin_id, Emp_id) → (requirement)

FD Closure : (Request_id, requirement_id, User_id, Admin_id, Emp_id) → (requirement)

Attribute Closure : None

20. Users (user_id, user_pwd, Fname, Lname, Location_address, village_id)

FD's : (User_id) → (user_pwd, Fname, Lname, Location_address, village_id)

FD Closure : (User_id) → (user_pwd), (User_id) → (Fname, Lname), (User_id) → (Location_address), (User_id) → (village_id)

Attribute Closure :

- (User_id)+ = (User_id, user_pwd, Fname, Lname, Location_address, village_id)

21. Employee_details (Emp_id, emp_pwd, emp_type, emp_age)

FD's : (Emp_id) → (emp_pwd, emp_type, emp_age)

FD Closure : (Emp_id) → (emp_pwd), (Emp_id) → (emp_type), (Emp_id) → (emp_age)

Attribute Closure :

- $(Emp_id)^+ = (Emp_id, emp_pwd, emp_type, emp_age)$

22.Employee_salary($Emp_id, emp_type, emp_salary$)

FD's : $(emp_type) \rightarrow (emp_salary)$

FD Closure : $(emp_type) \rightarrow (emp_salary)$

Attribute Closure :

- $(emp_type)^+ = (emp_type, emp_salary)$

23.Administrator($Admin_id, Request_id, requirement_id, project_status$)

FD's : $(Admin_id, Request_id, requirement_id) \rightarrow (project_status)$

FD Closure : $(Admin_id, Request_id, requirement_id) \rightarrow (project_status)$

Attribute Closure : None

24.Administrator($Admin_id, admin_pwd$)

FD's : $(Admin_id) \rightarrow (admin_pwd)$

FD Closure : $(Admin_id) \rightarrow (admin_pwd)$

Attribute Closure :

- $(Admin_id)^+ = (Admin_id, admin_pwd)$

After normalization, we have reduced most dependencies, as a result for each table the normal functional dependency matches with the FD Closure with some additional FD's but we could not derive more, because, most tables have one or two FD

Canonical Cover :-

A canonical cover (F_c) of a set of functional dependencies F is a simplified set of functional dependencies that has the same closure as the original set F .

So, no functional dependency in F_c contains an extraneous attribute.

Also, each left side of a FD, in F_c is unique.

Rules for reducing the extraneous attributes in the FD's:

In our project, since we have reduced the tables through normalization upto BCNF, there are no extraneous attributes. So, let's assume that the table/relation has some extraneous attributes, and try to find out canonical cover and prove that our assumption is wrong.

Let's take : Ex 1

Village_Details (village_id, village_name, district_name)

FD's : (village_id) \rightarrow (village_name, district_name)

FD Closure : (village_id) \rightarrow (village_name), (village_id) \rightarrow (district_name).

Attribute Closure :

- (village_id)⁺ = (village_id, village_name, district_name)

Let's assume village_name is extraneous in this table, it is RHS, so let's apply rule 2 for checking. So the new fd, $F' = \{ (village_id) \rightarrow (district_name) \}$. With this new FD we check if $\{ village_id \rightarrow village_name \}$ can be inferred from F' . To do so compute Attribute closure of village_id, if Attribute closure includes village_name, then village_name is extraneous in RHS. but in this case, village_id could not infer village_name from F' .

So from this, we could conclude that village_name is not extraneous in RHS.

Ex 2

Employee_details (Emp_id, emp_pwd, emp_type, emp_age)

FD's : (Emp_id) \rightarrow (emp_pwd, emp_type, emp_age)

FD Closure : (Emp_id) \rightarrow (emp_pwd), (Emp_id) \rightarrow (emp_type),

(Emp_id) \rightarrow (emp_age)

Attribute Closure :

- (Emp_id)⁺=(Emp_id, emp_pwd, emp_type, emp_age)

Now this table does not contain any extraneous attribute, so we try adding any extraneous attribute and try to reduce it through canonical cover. We add new attribute 'emp_DOB' with additional functional dependency { (Emp_id) \rightarrow (emp_DOB), (emp_DOB) \rightarrow (emp_age) }. So we know that by adding 'emp_DOB' it causes 'emp_age' to be extraneous, we will prove it through canonical cover.

Now new Fd, Fd closure and attribute closure,

Employee_details (Emp_id, emp_pwd, emp_type, emp_age, emp_DOB)

FD's : (Emp_id) \rightarrow (emp_pwd, emp_type, emp_age, emp_DOB), (emp_DOB) \rightarrow (emp_age)

FD Closure : (Emp_id) \rightarrow (emp_pwd), (Emp_id) \rightarrow (emp_type),

(Emp_id) \rightarrow (emp_age), (Emp_id) \rightarrow (emp_DOB)

Attribute Closure :

- (Emp_id)⁺=(Emp_id, emp_pwd, emp_type, emp_age, emp_DOB)

So, let's prove that emp_age is extraneous in this table by using rule 2.

F' = { (emp_DOB) \rightarrow (emp_age), (Emp_id) \rightarrow (emp_pwd, emp_type, emp_DOB) }

To check if $\text{Emp_id} \rightarrow \text{emp_age}$ can be inferred from F' , we compute $(\text{Emp_id})^+$

Under F' , and check if (Emp_id) includes emp_age , if so then emp_age is extraneous.

$(\text{Emp_id})^+ = (\text{Emp_id}, \text{emp_pwd}, \text{emp_type}, \text{emp_DOB}, \text{emp_age})$

So, Emp_id include emp_age . So emp_age is extraneous, Proved.

c)Anomalies:

Insertion anomaly:

The details or values for any devices can be entered only when the person is registered with that device.

Updation anomaly:

When the details about the user is updated or changed, the details should also be updated with all the devices the user is associated with or with all the devices the person owns. Also when the details about the device are updated the values or the details should also be updated with the user who owns or is associated with the device.

Deletion anomaly:

When the details about the user is deleted the details of the respective devices the user owns or is associated with also gets deleted. When the user owns only one device when the details about the user gets deleted, the details of the user also gets deleted.

3)First normal form:

1 NF :-

A. Identifying primary key attributes:

(village_id, village_name, district_name,

Source_id, Source_name, Source_capacity, Maintenance_review_dates, Village_id,

filtration_id, Filtration_method, filtration_location, filtration_capacity, Maintainance_review_date, ref_id,

chloride, fluoride, arsenic, grade_of_contamination, ref_id, filtration_id, improvement_index,

Bill_id, Tax_generated, funds, village_id, Connection_id, Connection_type,

Flat_rate, no_of_working_connections, no_of_out_of_connections, Village_id, connection_id,

length_of_pipeline, year_of_laying, pipe_type, sanction_id, deposit_works, village_id, Sanction_id,

sanctioned_amount, year_of_sanction, completion_year, Works_id,

sanction_id,max_transportation_of_water, IHHL_id, village_id, sanctioned_ihhls,inprogress_IHHL, Completed_IHHL, sanction_amt IHHL_fund_id,Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, IHHL_id, Request_id,requirement_id, user_id, village_id, requirement_id, user_id, requirement,request_type, User_id, user_pwd, Fname, Lname, Location_address, village_id,Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type,emp_age)

B. Removing repeating groups/ attributes

(village_id, village_name, district_name, Source_id, Source_name,Source_capacity, Maintenance_review_dates, filtration_id, Filtration_method,filtration_location, filtration_capacity, ref_id, chloride, fluoride, arsenic,, improvement_index, Bill_id, Tax_generated, funds,Connection_id, Connection_type, Flat_rate, no_of_working_connections,no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type,Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction,completion_year, Works_id, max_transportation_of_water, IHHL_id,inprogress_IHHL, Completed_IHHL, sanction_amt, IHHL_fund_id,Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, Request_id,requirement_id, requirement, request_type, User_id, user_pwd, Fname, Lname,Location_address, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd,emp_salary, emp_type, emp_age)

C. Identifying Multivalued/ Non-Atomic attributes

(village_id, village_name, district_name, Source_id, Source_name,Source_capacity, Maintenance_review_dates, filtration_id, Filtration_method,filtration_location, filtration_capacity, ref_id, chloride, fluoride, arsenic,grade_of_contamination, improvement_index, Bill_id, Tax_generated, funds,Connection_id, Connection_type, Flat_rate, no_of_working_connections,no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type,Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction,completion_year, Works_id, max_transportation_of_water, IHHL_id,inprogress_IHHL, Completed_IHHL, sanction_amt, IHHL_fund_id,Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, Request_id, requirement_id,requirement, request_type, User_id, user_pwd, Fname, Lname, Location_address,Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type,emp_age)

D. Decomposing based on business rules:

Rural Water supply:

(village_id, village_name, district_name, Source_id, Source_name, Source_capacity, Maintenance_review_dates, filtration_id, Filtration_method, filtration_location, filtration_capacity, ref_id, chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id, Tax_generated, funds, Connection_id, Connection_type, Flat_rate, no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

IHHL:

(IHHL_id, inprogress_IHHL, Completed_IHHL, sanction_amt, IHHL_fund_id, Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, village_id)

Users and requests:

(Request_id, requirement_id, requirement, request_type, User_id, user_pwd, Fname, Lname, Location_address, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age, village_id)

E. Decomposition of the table to separate out multivalued and non-atomic

attributes.

a. Rural water supply

i. Rural_water (village_id, village_name, district_name, Source_id, Source_name, Source_capacity, filtration_id, filtration_location, filtration_capacity, ref_id, chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id, Tax_generated, funds, Connection_id, Flat_rate, no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type,

Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction, completion_year,
Works_id, max_transportation_of_water)

ii. Water_sourcedetails (Source_id, Source_name, Source_capacity,
Maintenance_review_start_date, Maintenance_review_end_date)

iii. Filtration_method (filtration_id, Filtration_method)

iv. WaterConnections (Connection_id, Connection_type, Flat_rate)

b. IHHL

i. IHHL(IHHL_id, inprogress_IHHL, Completed_IHHL, sanction_amt, IHHL_fund_id,
Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, village_id)

c. Users and requests

i. users_and_requests(Request_id, requirement_id, requirement, request_type, User_id, user_pwd, Fname,
Lname, Location_address, Admin_id, admin_pwd, Emp_id, emp_pwd, emp_salary, emp_type, emp_age,
village_id)

ii. Admin(Admin_id, Request_id, requirement_id, project_status)

After reducing to 1-NF, we now have 7 tables.

4) Identifying the partial identities and checking for 2nd normalization form.

1) It is present in 1st normal form

2) Checking for Partial dependency:

Primary keys of each table :

1. Rural_water(village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, Works_id)

2. Water_sourcedetails (Source_id)
3. Filtration_method (filtration_id)
4. WaterConnections (Connection_id, Connection_type)
5. IHHL(IHHL_id, IHHL_fund_id)
6. users_and requests(Request_id, requirement_id, User_id, Admin_id,Emp_id)
7. Admin(Admin_id, Request_id, requirement_id)

Functional Dependencies :

I. Full FD's :

○ Rural_water:

1. None

○ Water_sourcedetails:

1. (Source_id) → (Source_name)
2. (Source_id) → (Source_capacity)
3. (Source_id) → (Maintenance_review_start_date)
4. (Source_id) → (Maintenance_review_end_date)

○ Filtration_method

1. (filtration_id) → (Filtration_method)

○ WaterConnections

1. (Connection_id,Connection_type) → (flat_rate)

○ IHHL

1. (IHHL_id, IHHL_fund_id) → (village_id)

○ users_and requests

1. (Request_id, requirement_id, User_id, Admin_id, Emp_id) →

(requirement)

○ Admin

1. (Admin_id, Request_id, requirement_id) → (project_status)

II. Partial FD's :

● Rural_water:

○ (village_id) → (village_name, district_name)

○ (Source_id) → (Source_name, Source_capacity,

Maintenance_review_start_date,

Maintenance_review_end_date)

○ (filtration_id) → (filtration_location, filtration_capacity)

○ (ref_id) → (chloride, fluoride, arsenic,

grade_of_contamination)

○ (Bill_id) → (Tax_generated, funds)

○ (Connection_id) → (no_of_working_connections,

no_of_out_of_connections)

○ (Connection_id) → (length_of_pipeline, year_of_laying,

pipe_type)

○ (Sanction_id) → (Sanctioned_amount, year_of_sanction, completion_year)

○ (Sanction_id) → (deposit_works)

○ (Works_id) → (max_transportation_of_water)

● Water_sourcedetails:

○ None

● Filtration_method

○ None

● WaterConnections

○ None

● IHHL

○ (IHHL_id) → (inprogress_IHHL, Completed_IHHL, sanction_amt)

○ (IHHL_fund_id) → (Sanctioned_IHHLs, Estimated_cost, funds_from_ngo)

● users_and requests

○ (Request_id, requirement_id) → (request_type)

○ (Request_id, requirement_id) → (village_id)

○ (User_id) → (user_pwd, Fname, Lname, Location_address, village_id)

- (Admin_id) → (admin_pwd)
- (Emp_id) → (emp_pwd, emp_type, emp_age)
- Admin
- None

Schema after 2nd normalization:

1. Rural_water(village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, Works_id)
2. Village_Details (village_id, village_name, district_name)
3. Water_source_details (Source_id, Source_name, Source_capacity, Maintenance_review_start_date, Maintenance_review_end_date)
4. Filtration_method (filtration_id, Filtration_method)
5. Filtration_details (filtration_id, filtration_location, filtration_capacity)
6. Quality_assurance (ref_id, chloride, fluoride, arsenic, grade_of_contamination)
7. Water_revenue_details (Bill_id, Tax_generated, funds)
8. Village_water_connections(Connection_id ,no_of_working_connections, no_of_out_of_connections)
9. Water_Connection_type (Connection_id, Connection_type, Flat_rate)
10. Pipeline_details (Connection_id, length_of_pipeline, year_of_laying,

pipe_type)

11.Sanction (Sanction_id, sanctioned_amount, year_of_sanction,
completion_year)

12.works (works_id, max_transportation_of_water)

13.Depositworks (Sanction_id, deposit_works)

14.village_IHHL (IHHL_id, IHHL_fund_id, village_id)

15.IHHL_expenditure_details (IHHL_fund_id, Sanctioned_IHHLs,
Estimated_cost, funds_from_ngo)

16.Individual_house_hold_latrines (IHHL_id, inprogress_IHHL,
Completed_IHHL, sanction_amt)

17.village_request(Request_id, requirement_id, village_id)

18.Requirements (Request_id, requirement_id, request_type)

19.User_requests(Request_id, requirement_id, User_id, Admin_id, Emp_id,
requirement)

20.Users (user_id, user_pwd, Fname, Lname, Location_address, village_id)

21.Employee_details (Emp_id, emp_pwd, emp_salary, emp_type, emp_age)

22.Administrator(Admin_id, Request_id, requirement_id, project_status)

23.Administrator(Admin_id, admin_pwd)

5)Identifying the transitive dependency and checking for 3rd normal form:

1) The given schema is in 2nd normal form

2) Check for transitive dependency:

Transitive dependencies :-

- (emp_type) → (emp_salary)

So decomposing these transitive decomposing gives us:

1. Employee_details (Emp_id, emp_pwd, emp_type, emp_age)
2. Employee_salary(Emp_id, emp_type, emp_salary)

Schema after 3rd normalization:

1. Rural_water(village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, Works_id)
2. Village_Details (village_id, village_name, district_name)
3. Water_source_details (Source_id, Source_name, Source_capacity, Maintenance_review_start_date, Maintenance_review_end_date)
4. Filtration_method (filtration_id, Filtration_method)
5. Filtration_details (filtration_id, filtration_location, filtration_capacity)
6. Quality_assurance (ref_id, chloride, fluoride, arsenic, grade_of_contamination)
7. Water_revenue_details (Bill_id, Tax_generated, funds)
8. Village_water_connections(Connection_id ,no_of_working_connections,

no_of_out_of_connections)

9. Water_Connection_type (Connection_id, Connection_type, Flat_rate)

10. Pipeline_details (Connection_id, length_of_pipeline, year_of_laying,
pipe_type)

11. Sanction (Sanction_id, sanctioned_amount, year_of_sanction,
completion_year)

12. works (works_id, max_transportation_of_water)

13. Depositworks (Sanction_id, deposit_works)

14. village_IHHL (IHHL_id, IHHL_fund_id, village_id)

15. IHHL_expenditure_details (IHHL_fund_id, Sanctioned_IHHLs,
Estimated_cost, funds_from_ngo)

16. Individual_house_hold_latrines (IHHL_id, inprogress_IHHL,
Completed_IHHL, sanction_amt)

17. village_request (Request_id, requirement_id, village_id)

18. Requirements (Request_id, requirement_id, request_type)

19. User_requests (Request_id, requirement_id, User_id, Admin_id, Emp_id,
requirement)

20. Users (user_id, user_pwd, Fname, Lname, Location_address, village_id)

21. Employee_details (Emp_id, emp_pwd, emp_type, emp_age)

22. Employee_salary (Emp_id, emp_type, emp_salary)

23.Administrator(Admin_id, Request_id, requirement_id, project_status)

24.Administrator(Admin_id, admin_pwd)

6)Lossless decomposition and dependency preserving:

_1. Rural Water Supply (village_id, village_name, district_name, Source_id, Source_name, Source_capacity, Maintenance_review_date, filtration_id,

Filtration_method , filtration_location, filtration_capacity, ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated, funds, Connection_id , Connection_type , Flat_rate, no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id,

max_transportation_of_water)

$F = \{(Source_id) \rightarrow (Source_name), (Source_id) \rightarrow (Source_capacity), (Source_id) \rightarrow (Maintenance_review_start_date), (Source_id) \rightarrow (Maintenance_review_end_date), (filtration_id) \rightarrow (Filtration_method), (Connection_id, Connection_type) \rightarrow (Flat_rate), (IHHL_id, IHHL_fund_id) \rightarrow (village_id), (village_id) \rightarrow (village_name, district_name), (Source_id) \rightarrow (Source_name, Source_capacity, Maintenance_review_start_date, Maintenance_review_end_date), (filtration_id) \rightarrow (filtration_location, filtration_capacity), (ref_id) \rightarrow (chloride, fluoride, arsenic, grade_of_contamination), (Bill_id) \rightarrow (Tax_generated, funds), (Connection_id) \rightarrow (no_of_working_connections, no_of_out_of_connections), (Connection_id) \rightarrow (length_of_pipeline, year_of_laying, pipe_type), (Sanction_id) \rightarrow (Sanctioned_amount, year_of_sanction, completion_year), (Sanction_id) \rightarrow (deposit_works), (Works_id) \rightarrow (max_transportation_of_water)\}$

is decomposed to

a. Rural_water (village_id, Source_id, Source_name, Source_capacity, Maintenance_review_date, filtration_id, Filtration_method , filtration_location, filtration_capacity, ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated, funds, Connection_id , Connection_type , Flat_rate, no_of_working_connections, no_of_out_of_connections,

length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

b. Village_Details (village_id, village_name, district_name) Rural_Water \square Village_Details
village_id Village_id is the key for village_details. (Lossless decomposition)

$$F1 = \{(village_id) \rightarrow (village_name, district_name)\}$$

2. Rural_water (village_id, Source_id, Source_name, Source_capacity, Maintenance_review_dates, filtration_id, Filtration_method , filtration_location, filtration_capacity, ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated, funds, Connection_id , Connection_type , Flat_rate, no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

is decomposed to

a. Rural_water (village_id, Source_id, filtration_id, Filtration_method filtration_location, filtration_capacity, ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated, funds, Connection_id , Connection_type , Flat_rate, no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

b. Water_source_details (Source_id, Source_name, Source_capacity, Maintenance_review_start_date, Maintenance_review_end_date)

Rural_water \square Water_source_details Source_id That is key for water_source_details. Therefore, lossless decomposition

$$F2 = \{(Source_id) \rightarrow (Source_name), (Source_id) \rightarrow (Source_capacity), (Source_id) \rightarrow (Maintenance_review_start_date), (Source_id) \rightarrow (Maintenance_review_end_date)\}$$

3. Rural_water (village_id, Source_id, filtration_id,

Filtration_method , filtration_location, filtration_capacity, ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated, funds, Connection_id ,

Connection_type , Flat_rate, no_of_working_connections, no_of_out_of_connections,
length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works, Sanctioned_amount,
year_of_sanction, completion_year, Works_id,

max_transportation_of_water) is decomposed to

a. Rural_water (village_id, Source_id, filtration_id,

ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated,
funds, Connection_id , Connection_type , Flat_rate, no_of_working_connections,
no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works,
Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

b. Filtration_method (filtration_id, Filtration_method)

c. Filtration_details (filtration_id, filtration_location, filtration_capacity)

Rural_water \square Filtration_method \square filtration_details \rightarrow filtration_id

That is filtration_id is key for filtration_details. Therefore, lossless decomposition

F3= {(filtration_id) \rightarrow (Filtration_method), (filtration_id) \rightarrow (filtration_location, filtration_capacity)}

4. Rural_water (village_id, Source_id, filtration_id,

ref_id , chloride, fluoride, arsenic, grade_of_contamination, improvement_index, Bill_id , Tax_generated,
funds, Connection_id , Connection_type , Flat_rate, no_of_working_connections,
no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works,
Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

is decomposed to

a. Rural_water (village_id, Source_id, filtration_id, ref_id , improvement_index, Bill_id ,
Tax_generated, funds, Connection_id , Connection_type, Flat_rate, no_of_working_connections,
no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type, Sanction_id , deposit_works,

Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)b.
 Quality_assurance (ref_id, chloride, fluoride,
 arsenic, grade_of_contamination)

Rural_water \square quality_assurance ref_id

That is key for quality assurance. Therefore, lossless decomposition

$F4 = \{(ref_id) \rightarrow (chloride, fluoride, arsenic, grade_of_contamination)\}$

5. Rural_water (village_id, Source_id, filtration_id,

ref_id, improvement_index, Bill_id, Tax_generated, funds, Connection_id, Connection_type, Flat_rate,
 no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type,
 Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id,
 max_transportation_of_water)is decomposed to

a. Rural_water (village_id, Source_id, filtration_id,

ref_id, improvement_index, Bill_id, Connection_id, Connection_type, Flat_rate,
 no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type,
 Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id,
 max_transportation_of_water)

b. Water_Revenue_Details (Bill_id, Tax_generated, funds)

Rural_water \square water_revenue_details \rightarrow bill_id That is key in water_revenue_details. Therefore, lossless decomposition

$F5 = \{(Bill_id) \rightarrow (Tax_generated, funds)\}$

6. Rural_water (village_id, Source_id, filtration_id,

ref_id, improvement_index, Bill_id, Connection_id, Connection_type, Flat_rate,
 no_of_working_connections, no_of_out_of_connections, length_of_pipeline, year_of_laying, pipe_type,

Sanction_id , deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

is decomposed to

a. Rural_water (village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

b. Village_water_connections (Connection_id, no_of_working_connections, no_of_out_of_connections)

c. Water_Connection_type (Connection_id, Connection_type, Flat_rate)

d. Pipeline_details (Connection_id, length_of_pipeline, year_of_laying, pipe_type)

Rural_water \square village_water_connections \square water_connection_type \square pipeline_details connection_id

That is the key for pipeline_details. Therefore, lossless decomposition

$F_6 = \{ (\text{Connection_id}, \text{Connection_type}) \rightarrow (\text{flat_rate}), (\text{Connection_id}) \rightarrow (\text{no_of_working_connections}, \text{no_of_out_of_connections}), (\text{Connection_id}) \rightarrow (\text{length_of_pipeline}, \text{year_of_laying}, \text{pipe_type}) \}$

7. Rural_water (village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, deposit_works, Sanctioned_amount, year_of_sanction, completion_year, Works_id, max_transportation_of_water)

Is decomposed to

a. Rural_water (village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id)

b. Sanction (Sanction_id, sanctioned_amount, year_of_sanction, completion_year)

c. Deposit_works (Sanction_id, deposit_works, max_transportation_of_water)

Rural_water \square sanction \square deposit_works sanction_id

That is key for deposit_works. Therefore, lossless decomposition

$F7 = \{(Sanction_id) \rightarrow (Sanctioned_amount, year_of_sanction, completion_year), (Sanction_id) \rightarrow (deposit_works), (Works_id) \rightarrow (max_transportation_of_water)\}$

$F = F1 \cup F2 \cup F3 \cup F4 \cup F5 \cup F6 \cup F7$

This condition is satisfied. This is Dependency Preserving.

8. IHHL (IHHL_id, inprogress_IHHL, Completed_IHHL, sanction_amt, IHHL_fund_id, Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, village_id)

Is decomposed to

- a. village_IHHL (IHHL_id, IHHL_fund_id, village_id)
- b. Individual_house_hold_latrines (IHHL_id, inprogress_IHHL, Completed_IHHL, sanction_amt)
- c. IHHL (IHHL_id, IHHL_fund_id, Sanctioned_IHHLs, Estimated_cost, funds_from_ngo, village_id)

Village_ihhl \square individual_house_hold_latrines \square ihhl IHHL_id

It is key for all the three. Therefore, lossless decomposition

9. IHHL (IHHL_id, IHHL_fund_id, Sanctioned_IHHLs,

Estimated_cost, funds_from_ngo, village_id) is decomposed to

- a. IHHL_expenditure_details (IHHL_fund_id, sanctioned_IHHLs, Estimated_cost, funds_from_ngo)
- b. IHHL (IHHL_id, IHHL_fund_id, village_id)

IHHL_Expenditure \square IHHL IHHL_Fund_id

It is key in IHHL_Expenditure_details. Therefore, lossless decomposition

10. Users and requests (Request_id, requirement_id, requirement, request_type, User_id, user_pwd, Fname, Lname, Location_address, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age, village_id)

Is decomposed into

a. village_request (Request_id, requirement_id, village_id)

b. Requirements (Request_id, requirement_id, request_type)

c. User_requests (Request_id, requirement_id, User_id, Admin_id Emp_id, requirement)

d. Users and requests (Request_id, requirement_id, User_id, user_pwd, Fname, Lname, Location_address, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age, village_id)

village_request \square Requirements \square User_requests \square Users and requests request_id, requirement_id

It is a key in village_request. Therefore, lossless decomposition.

11. Users and requests (Request_id, requirement_id, User_id, user_pwd, Fname, Lname, Location_address, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age, village_id)

Is decomposed into

a. Users (user_id, user_pwd, Fname, Lname, Location_address, village_id)

b. Users and requests (Request_id, requirement_id, User_id, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age, village_id)

Users \square Users_and_requests user_id

That is a key in Users. Therefore, lossless decomposition

12. Users and requests (Request_id, requirement_id, User_id, Admin_id, admin_pwd, project_status, Emp_id, emp_pwd, emp_salary, emp_type, emp_age, village_id)

Is decomposed into

- a. Employee_details (Emp_id, emp_pwd, emp_type, emp_age)
- b. Employee salary (Emp_id, emp_type, emp_salary)
- c. Users and requests (Request_id, requirement_id, User_id, Admin_id, admin_pwd, project_status, Emp_id, village_id)

Employee_details \square Employee_salary \square Users and requests Emp_id

That is a key in Employee_details. Therefore, lossless decomposition

13. Users and requests (Request_id, requirement_id, User_id, Admin_id, admin_pwd, project_status, Emp_id, village_id)

Is decomposed into

- a. Administrator_status (Admin_id, Request_id, requirement_id, project_status)
- b. Administrator (Admin_id, admin_pwd)
- c. User_requests (Request_id, requirement_id, User_id, Admin_id, Emp_id, requirement)

Administrator_status \square Administrator \square User_requests admin_id

That is the key for Administrator. Therefore, lossless decomposition

7)Normalise to bcnf if possible:

A table complies with BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X should be the super key of the table. In our project, after reducing 3-NF, all the tables satisfies this condition, as a result the tables are in BCNF.

Available functional dependencies:-

1. (Source_id) → (Source_name)
2. (Source_id) → (Source_capacity)
3. (Source_id) → (Maintenance_review_start_date)
4. (Source_id) → (Maintenance_review_end_date)
5. (filtration_id) → (Filtration_method)
6. (Connection_id, Connection_type) → (flat_rate)
7. (IHHL_id, IHHL_fund_id) → (village_id)
8. (Request_id, requirement_id, User_id, Admin_id, Emp_id) → (requirement)
9. (Admin_id, Request_id, requirement_id) → (project_status)
10. (village_id) → (village_name, district_name)
11. (Source_id) → (Source_name, Source_capacity, Maintenance_review_start_date, Maintenance_review_end_date)
12. (filtration_id) → (filtration_location, filtration_capacity)
13. (ref_id) → (chloride, fluoride, arsenic, grade_of_contamination)
14. (Bill_id) → (Tax_generated, funds)
15. (Connection_id) → (no_of_working_connections, no_of_out_of_connections)
16. (Connection_id) → (length_of_pipeline, year_of_laying, pipe_type)

17.(Sanction_id) → (Sanctioned_amount, year_of_sanction, completion_year)

18.(Sanction_id) → (deposit_works)

19.(Works_id) → (max_transportation_of_water)

20.(IHHL_id) → (inprogress_IHHL, Completed_IHHL, sanction_amt)

21.(IHHL_fund_id) → (Sanctioned_IHHLs, Estimated_cost, funds_from_ngo)

22.(Request_id, requirement_id) → (request_type)

23.(Request_id, requirement_id) → (village_id)

24.(User_id) → (user_pwd, Fname, Lname, Location_address, village_id)

25.(Admin_id) → (admin_pwd)

26.(Emp_id) → (emp_pwd, emp_type, emp_age)

27.(emp_type) → (emp_salary)

Data dependency table:

1. Rural_water(village_id, Source_id, filtration_id, ref_id, Bill_id, Connection_id, Sanction_id, Works_id)

2. Village_Details (village_id, village_name, district_name)

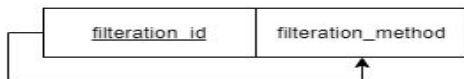


3. Water_source_details (Source_id, Source_name, Source_capacity, Maintenance_review_start_date, Maintenance_review_end_date)

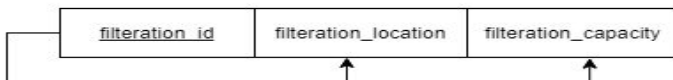


4. Filtration_method (filtration_id, Filtration_method)

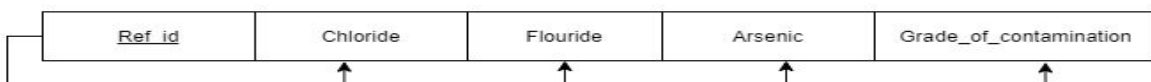
4. Filtration_method (filtration_id, Filtration_method)



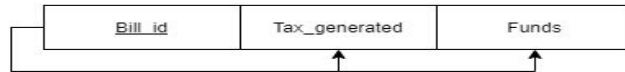
5. Filtration_details (filtration_id, filtration_location, filtration_capacity)



6. Quality_assurance (ref_id, chloride, fluoride, arsenic, grade_of_contamination)



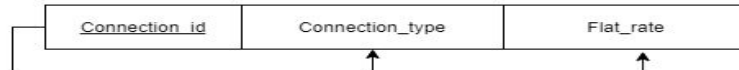
7. Water_revenue_details (Bill_id, Tax_generated, funds)



8. Village_water_connections(Connection_id,no_of_working_connections, no_of_out_of_connections)



9. Water_Connection_type (Connection_id, Connection_type, Flat_rate)



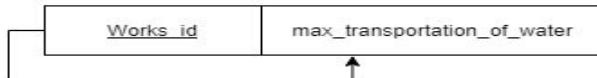
10. Pipeline_details (Connection_id, length_of_pipeline, year_of_laying, pipe_type)



11. Sanction (Sanction_id, sanctioned_amount, year_of_sanction, completion_year)



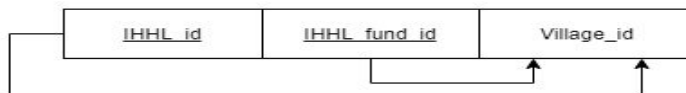
12. works (works_id, max_transportation_of_water)



13. Depositworks (Sanction_id, deposit_works)



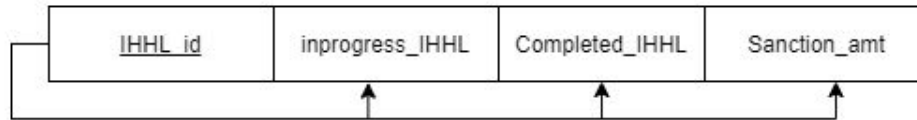
14. village_IHHL (IHHL_id, IHHL_fund_id, village_id)



15. IHHL_expenditure_details (IHHL_fund_id, Sanctioned_IHHLs, Estimated_cost, funds_from_ngo)

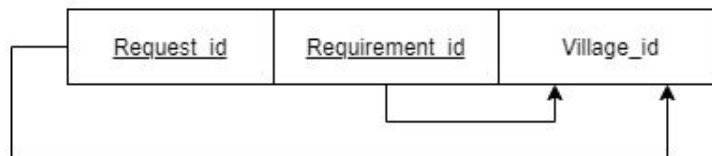


16.Individual_house_hold_latrines (IHHL_id, inprogress_IHHL, Completed_IHHL, sanction_amt)

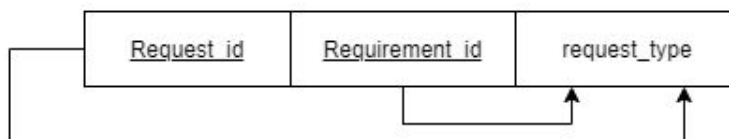


17.village_request(Request_id, requirement_id, village_id)

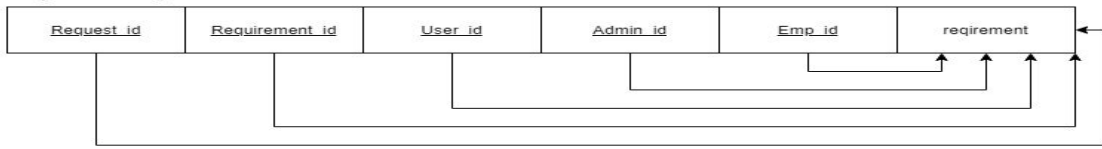
|



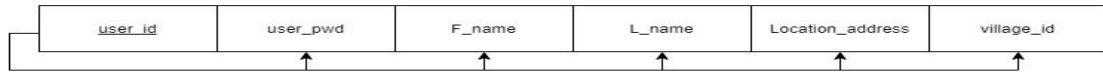
18.Requirements (Request_id, requirement_id, request_type)



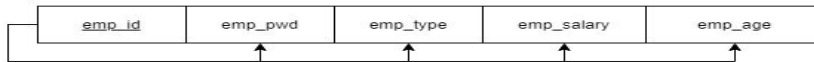
19. User_requests(Request_id, requirement_id, User_id, Admin_id, Emp_id, requirement)



20. Users (user_id, user_pwd, Fname, Lname, Location_address, village_id)



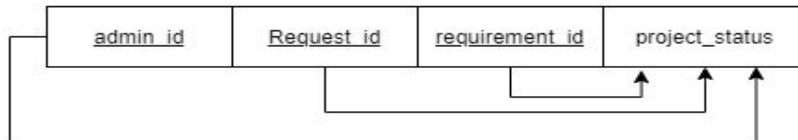
21. Employee_details (Emp_id, emp_pwd, emp_type, emp_age)



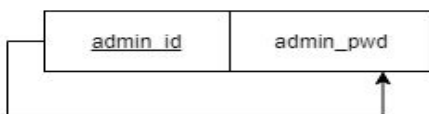
22. Employee_salary(Emp_id, emp_type, emp_salary)



23. Administrator(Admin_id, Request_id, requirement_id, project_status)



24. Administrator_login(Admin_id, admin_pwd)



Chapter 7 Backend design

Table creation comments:

We used a NoSQL design and we structured the DB in a way that we categorised which entities and attributes are important and then, created the DB design with the inputs from the normalised schema of the SQL DB. Reasons for using a NoSQL DB over an SQL based ones is that, while developing an app latency and response times are important factors and using a NoSQL based structure helps achieving these factors easily.

Sample instances of the tables:

Creation of Tables

- a. Include the create command for ALL THE TABLES, sample insert commands.
- b. Minimum 5 meaningful records to be inserted.
- c. Data from all the tables.

```
INSERT INTO public."Rural_water"(village_id, "Source_id", filtration_id, ref_id, "Bill_id",  
"Connection_id", "Sanction_id", "Works_id")
```

```
VALUES (101,201,301,401, 501, 601, 701, 801);
```

```
INSERT INTO public."Rural_water"(village_id, "Source_id", filtration_id, ref_id, "Bill_id",  
"Connection_id", "Sanction_id", "Works_id")
```

```
VALUES (102,202,302,402, 502, 602, 702, 802);
```

```
INSERT INTO public."Rural_water"(  
village_id, "Source_id", filtration_id, ref_id, "Bill_id", "Connection_id", "Sanction_id", "Works_id")
```

```
VALUES (103,203,303,403, 503, 603, 703, 803);
```

```
VALUES (103,203,303,403, 503, 603, 703, 803);
```

```
INSERT INTO public."Rural_water"(  
village_id, "Source_id", filtration_id, ref_id, "Bill_id", "Connection_id", "Sanction_id", "Works_id")
```

```
VALUES (104,204,304,404, 504, 604, 704, 804);
```

```
VALUES (104,204,304,404, 504, 604, 704, 804);
```

```
INSERT INTO public."Rural_water"(  
village_id, "Source_id", filtration_id, ref_id, "Bill_id", "Connection_id", "Sanction_id", "Works_id")
```

```
VALUES (105,205,305,405, 505, 605, 705, 805);
```

```
VALUES (105,205,305,405, 505, 605, 705, 805);
```

Data Output									Explain	Messages	Notifications
	<u>village_id</u> [PK] integer	<u>Source_id</u> [PK] integer	<u>litration_id</u> [PK] integer	<u>ref_id</u> [PK] integer	<u>Bill_id</u> [PK] integer	<u>Connection_id</u> [PK] integer	<u>Sanction_id</u> [PK] integer	<u>Works_id</u> [PK] integer			
1	101	201	301	401	501	601	701	801			
2	102	202	302	402	502	602	702	802			
3	103	203	303	403	503	603	703	803			
4	104	204	304	404	504	604	704	804			
5	105	205	305	405	505	605	705	805			

```
INSERT INTO public."Village_details"(village_id, village_name, district_name)
```

```
VALUES (101, 'lepakshi', 'ananthapuram');
```

```
INSERT INTO public."Village_details"(village_id, village_name, district_name)
```

```
VALUES (102, 'neelakunta', 'chittoor');
```

```
INSERT INTO public."Village_details"(village_id, village_name, district_name)
```

```
VALUES (103, 'rajolu', 'eastgodavari');
```

```
INSERT INTO public."Village_details"(village_id, village_name, district_name)
```

```
VALUES (104, 'chamaru', 'guntur');
```

```
INSERT INTO public."Village_details"(village_id, village_name, district_name)
```

```
VALUES (105, 'duvva', 'westgodavari');
```

village_id [PK] integer	village_name character varying (50)	district_name character varying (50)
1	101 lepakshi	ananthapuram
2	102 neelakunta	chittoor
3	103 rajolu	eastgodavari
4	104 chamaru	guntur
5	105 duvva	westgodavari

```
INSERT INTO public."Water_source_details"("Source_id", "Source_name", "Source_capacity",
"Maintenance_review_start_date", "Maintenance_review_end_date")
```

```
VALUES (201, 'ramarao_pond', 3500, '2_sep_2017', '14_sep_2017');
```

```
INSERT INTO public."Water_source_details"("Source_id", "Source_name", "Source_capacity",
"Maintenance_review_start_date", "Maintenance_review_end_date")
```

```
VALUES (202, 'krishnayya_pond', 5840, '17-aug-17', '24-aug-17');
```

```
INSERT INTO public."Water_source_details"("Source_id", "Source_name", "Source_capacity",
"Maintenance_review_start_date", "Maintenance_review_end_date")
```

```
VALUES (203, 'durgarao_pond', 3400, '14-may-17', '22-may-17');
```

```
INSERT INTO public."Water_source_details"("Source_id", "Source_name", "Source_capacity",
"Maintenance_review_start_date", "Maintenance_review_end_date")
```

```
VALUES (204, 'durgamma_pond', 5400, '16-jul-17', '27-jul-17');
```

```
INSERT INTO public."Water_source_details"("Source_id", "Source_name", "Source_capacity", "Maintenance_review_start_date",
```

```
"Maintenance_review_end_date")
```

```
VALUES (205, 'ranggayya_pond', 4800, '04-feb-17', '19-feb-17');
```

Data Output	Explain	Messages	Notifications
Source_id [PK] integer	Source_name character varying (30)	Source_capacity integer	Maintenance_review_start_date character varying (30) Maintenance_review_end_date character varying (30)
1	201 namarao_pond	3500	14_sep_2017
2	202 krishnaya_pond	5840	17 aug 17
3	203 durgarao_pond	3400	14 may 17
4	204 durgamma_pond	5400	16 jul 17
5	205 ranggayya_pond	4800	04 feb 17

```
INSERT INTO public." Filtration_method"(filtration_id, filtration_method)VALUES (301, 'slow sand
filter');
```

```
INSERT INTO public." Filtration_method"(filtration_id, filtration_method)VALUES (301, 'rapid sand
filter ');
```

```
INSERT INTO public." Filtration_method"(filtration_id, filtration_method)VALUES (302, ' slow sand
filter');
```

```
INSERT INTO public." Filtration_method"(filtration_id, filtration_method)VALUES (303, 'rapid sand
filter');
```

Data Output	Explain	Messages	Notifications
filtration_id [PK] integer	filtration_method [PK] character varying		
1	301 rapid sand filter		
2	301 slow sand filter		
3	302 slow sand filter		
4	303 rapid sand filter		
5	304 slow sand filter		

```
INSERT INTO public." Filtration_method"(filtration_id, filtration_method)VALUES (304, 'slow sand
filter');
```

```
INSERT INTO public."Filtration_details"(filtration_id, filtration_location, filtration_capacity)VALUES
(301, 'pathapeta', 5);
```

```
INSERT INTO public."Filtration_details"(filtration_id, filtration_location, filtration_capacity)VALUES
(302, 'rajula colony', 7);
```

```
INSERT INTO public."Filtration_details"(filtration_id, filtration_location, filtration_capacity)VALUES
(303, 'subbarao thota', 4);
```

```
INSERT INTO public."Filtration_details"(filtration_id, filtration_location, filtration_capacity)VALUES
(304, 'kotadibba', 3);
```

```
INSERT INTO public."Filtration_details"(filtration_id, filtration_location, filtration_capacity)VALUES
(305, 'iddiramma colony', 5);
```



filtration_id	filtration_location	filtration_capacity
301	pathapeta	5
302	rajula colony	7
303	subbarao thota	4
304	kotadibba	3
305	iddiramma colony	5

```
INSERT INTO public."Quality_assurance"(ref_id, chloride, fluoride, arsenic,
grade_of_contamination)VALUES (401, 0.5, 0.6, 0.2,'a' );
```

```
INSERT INTO public."Quality_assurance"(ref_id, chloride, fluoride, arsenic,
grade_of_contamination)VALUES (402, 0.4, 0.5, 0.3, 'b');
```

```
INSERT INTO public."Quality_assurance"(ref_id, chloride, fluoride, arsenic,
grade_of_contamination)VALUES (403, 0.6, 0.4, 0.4, 'c');
```

```
INSERT INTO public."Quality_assurance"(ref_id, chloride, fluoride, arsenic,
grade_of_contamination)VALUES (404, 0.8, 0.2, 0.6, 'a');
```

```
INSERT INTO public."Quality_assurance"(ref_id, chloride, fluoride, arsenic,
grade_of_contamination)VALUES (405, 0.4, 0.6, 0.6, 'c');
```

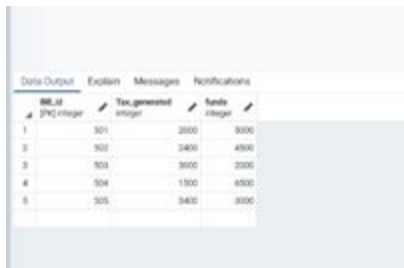


ref_id	chloride	fluoride	arsenic	grade_of_contamination
401	0.5	0.6	0.2	a
402	0.4	0.5	0.3	b
403	0.6	0.4	0.4	c
404	0.8	0.2	0.6	a
405	0.4	0.6	0.6	c

```

INSERT INTO public."Water_revenue_details"("Bill_id", "Tax_generated", funds)VALUES (501, 2000, 5000);
INSERT INTO public."Water_revenue_details"("Bill_id", "Tax_generated", funds)VALUES (502, 2400, 4500);
INSERT INTO public."Water_revenue_details"("Bill_id", "Tax_generated", funds)VALUES (503, 3000, 2000);
INSERT INTO public."Water_revenue_details"("Bill_id", "Tax_generated", funds)VALUES (504, 1500, 6500);
INSERT INTO public."Water_revenue_details"("Bill_id", "Tax_generated", funds)VALUES (505, 3400, 3000);

```



	Bill_id integer	Tax_generated integer	funds integer
1	501	2000	5000
2	502	2400	4500
3	503	3000	2000
4	504	1500	6500
5	505	3400	3000

```

INSERT INTO public."Village_water_connections"("Connection_id", no_of_working_connections, no_of_out_of_connections)VALUES (601, 4, 3);
INSERT INTO public."Village_water_connections"("Connection_id", no_of_working_connections, no_of_out_of_connections)VALUES (602, 5, 1);
INSERT INTO public."Village_water_connections"("Connection_id", no_of_working_connections, no_of_out_of_connections)VALUES (603, 2, 1);
INSERT INTO public."Village_water_connections"("Connection_id", no_of_working_connections, no_of_out_of_connections)VALUES (604, 6, 3);
INSERT INTO public."Village_water_connections"("Connection_id", no_of_working_connections, no_of_out_of_connections)VALUES (605, 6, 2);

```



	Connection_id integer	no_of_working_connections integer	no_of_out_of_connections integer
1	601	4	3
2	602	5	1
3	603	2	1
4	604	6	3
5	605	6	2

```

INSERT INTO public." Water_Connection_type"("Connection_id", "Connection_type", "Flat_rate")
VALUES (601, 'domestic metered', 5000);
INSERT INTO public." Water_Connection_type"("Connection_id", "Connection_type", "Flat_rate")
VALUES (602, 'commercial metered', 6000);
INSERT INTO public." Water_Connection_type"("Connection_id", "Connection_type", "Flat_rate")
VALUES (603, 'industrial metered', 8000);

```

```

INSERT INTO public."Water_Connection_type"("Connection_id", "Connection_type", "Flat_rate")
VALUES (604, 'commercial metered', 6000);
INSERT INTO public."Water_Connection_type"("Connection_id", "Connection_type", "Flat_rate")
VALUES (605, 'industrial metered', 8000);

```

Data Output	Explain	Messages	Notifications
Connection_id (PK) integer	Connection_type (PK) character varying (20)	Flat_rate integer	
1	601 Domestic metered	3000	
2	602 Commercial metered	6000	
3	603 industrial metered	8000	
4	604 Commercial metered	6000	
5	605 industrial metered	8000	

```

INSERT INTO public."Sanction"("Sanction_id", sanctioned_amount, year_of_sanction,
completion_year)
VALUES (701, 25000, 2016, 2017);
INSERT INTO public."Sanction"("Sanction_id", sanctioned_amount, year_of_sanction,
completion_year)
VALUES (702, 30000, 2017, 2018);
INSERT INTO public."Sanction"("Sanction_id", sanctioned_amount, year_of_sanction,
completion_year)
VALUES (703, 75000, 2016, 2017);
INSERT INTO public."Sanction"("Sanction_id", sanctioned_amount, year_of_sanction,
completion_year)
VALUES (704, 125000, 2018, 2019);
INSERT INTO public."Sanction"("Sanction_id", sanctioned_amount, year_of_sanction,
completion_year)
VALUES (705, 95000, 2015, 2016);

```

Data Output	Explain	Messages	Notifications
Connection_id (PK) integer	Connection_type (PK) character varying (20)	Flat_rate integer	
1	601 Domestic metered	3000	
2	602 Commercial metered	6000	
3	603 industrial metered	8000	
4	604 Commercial metered	6000	
5	605 industrial metered	8000	

```

INSERT INTO public."Pipeline_details"("Connection_id", length_of_pipeline, year_of_laying,
pipe_type)VALUES (601, 500, 2017, 'pvc pipeline');
INSERT INTO public."Pipeline_details"("Connection_id", length_of_pipeline, year_of_laying,
pipe_type)VALUES (602, 1000, 2016, 'copper pipeline');
INSERT INTO public."Pipeline_details"("Connection_id", length_of_pipeline, year_of_laying,
pipe_type)VALUES (603, 300, 2018, 'cast iron pipeline');
INSERT INTO public."Pipeline_details"("Connection_id", length_of_pipeline, year_of_laying,
pipe_type)VALUES (604, 800, 2017, 'copper pipeline');

```

```
INSERT INTO public."Pipeline_details"("Connection_id", length_of_pipeline, year_of_laying,
pipe_type)VALUES (605, 450, 2019, ' pvc pipeline');
```

	Connection_id	length_of_pipeline	year_of_laying	pipe_type
	[PK] integer	integer	integer	character varying (30)
1	601	500	2017	pvc pipeline
2	602	1000	2016	copper pipeline
3	603	300	2018	cast iron pipeline
4	604	800	2017	copper pipeline
5	605	400	2019	pvc pipeline

```
INSERT INTO public.works(works_id, max_transportation_of_water)VALUES (801, 200);
INSERT INTO public.works(works_id, max_transportation_of_water)VALUES (802, 250);
INSERT INTO public.works(works_id, max_transportation_of_water)VALUES (803, 400);
INSERT INTO public.works(works_id, max_transportation_of_water)VALUES (804, 320);
INSERT INTO public.works(works_id, max_transportation_of_water)VALUES (805, 290);
```

	works_id	max_transportation_of_water
	[PK] integer	integer
1	801	200
2	802	250
3	803	400
4	804	320
5	805	290

```
INSERT INTO public."Depositworks"("Sanctioned_id", deposit_works)VALUES (701, 801);
INSERT INTO public."Depositworks"("Sanctioned_id", deposit_works)VALUES (702, 802);
INSERT INTO public."Depositworks"("Sanctioned_id", deposit_works)VALUES (703, 803);
INSERT INTO public."Depositworks"("Sanctioned_id", deposit_works)VALUES (704, 804);
INSERT INTO public."Depositworks"("Sanctioned_id", deposit_works)VALUES (705, 805);
```

	Sanctioned_id	deposit_works
	[PK] integer	character varying
1	701	801
2	702	802
3	703	803
4	704	804
5	705	805

```

INSERT INTO public."village_IHHL"("IHHL_id", "IHHL_fund_id", village_id)VALUES (901,1001 ,
101);
INSERT INTO public."village_IHHL"("IHHL_id", "IHHL_fund_id", village_id)VALUES (902 ,1002 ,
102);
INSERT INTO public."village_IHHL"("IHHL_id", "IHHL_fund_id", village_id)VALUES (903,1003 ,
103);
INSERT INTO public."village_IHHL"("IHHL_id", "IHHL_fund_id", village_id)VALUES (904,1004 ,
104);
INSERT INTO public."village_IHHL"("IHHL_id", "IHHL_fund_id", village_id)VALUES (905,1005 ,
105);






```

	Data Output	Explain	Messages	Notifications
	IHHL_id [PK] integer		IHHL_fund_id [PK] integer	village_id integer
1	901		1001	101
2	902		1002	102
3	903		1003	103
4	904		1004	104
5	905		1005	105

```

INSERT INTO public."IHHL_expenditure_details"("IHHL_fund_id", "Sanctioned_IHHLs",
"Estimated_cost", funds_from_ngo)VALUES (1001, 2, 40000, 20000);
INSERT INTO public."IHHL_expenditure_details"("IHHL_fund_id", "Sanctioned_IHHLs",
"Estimated_cost", funds_from_ngo)VALUES (1002, 4, 80000, 40000);
INSERT INTO public."IHHL_expenditure_details"("IHHL_fund_id", "Sanctioned_IHHLs",
"Estimated_cost", funds_from_ngo)VALUES (1003, 3, 60000, 20000);
INSERT INTO public."IHHL_expenditure_details"("IHHL_fund_id", "Sanctioned_IHHLs",
"Estimated_cost", funds_from_ngo)VALUES (1004, 5, 100000, 60000);
INSERT INTO public."IHHL_expenditure_details"("IHHL_fund_id", "Sanctioned_IHHLs",
"Estimated_cost", funds_from_ngo)VALUES (1005, 4, 80000, 40000);






```

Data Output		Explain	Messages	Notifications	
	IHHL_fund_id [PK] integer 	Sanctioned_IHHLs character varying 	Estimated_cost integer 	funds_from_ngo integer 	
1	1001	2	40000	20000	
2	1002	4	80000	40000	
3	1003	3	60000	20000	
4	1004	5	100000	60000	
5	1005	4	80000	40000	


```

INSERT INTO public."Individual_house_hold_latrines"("IHHL_id", "inprogress_IHHL",
"Completed_IHHL", sanction_amt)VALUES (901, 2, 1, 20000);
INSERT INTO public."Individual_house_hold_latrines"("IHHL_id", "inprogress_IHHL",
"Completed_IHHL", sanction_amt)VALUES (901, 4, 2, 40000);
INSERT INTO public."Individual_house_hold_latrines"("IHHL_id", "inprogress_IHHL",
"Completed_IHHL", sanction_amt)VALUES (901, 3, 1, 30000);
INSERT INTO public."Individual_house_hold_latrines"("IHHL_id", "inprogress_IHHL",
"Completed_IHHL", sanction_amt)VALUES (901, 2, 3, 40000);
INSERT INTO public."Individual_house_hold_latrines"("IHHL_id", "inprogress_IHHL",
"Completed_IHHL", sanction_amt)VALUES (905, 2, 2, 30000);

```

Data Output		Explain	Messages	Notifications
	IHHL_id [PK] integer 	inprogress_IHHL character varying 	Completed_IHHL character varying 	sanction_amt integer 
1	901	2	1	20000
2	902	4	2	40000
3	903	3	1	30000
4	904	2	3	40000
5	905	2	2	30000

```

INSERT INTO public.village_request("Request_id", requirement_id, village_id)VALUES (1101, 1201,
101);
INSERT INTO public.village_request("Request_id", requirement_id, village_id)VALUES (1102, 1202,
102);
INSERT INTO public.village_request("Request_id", requirement_id, village_id)VALUES (1103, 1203,
103);
INSERT INTO public.village_request("Request_id", requirement_id, village_id)VALUES (1104, 1204,
104);
INSERT INTO public.village_request("Request_id", requirement_id, village_id)VALUES (1105, 1205,
105);

```

Data Output	Explain	Messages	Notifications
Request_id [PK] integer	requirement_id [PK] integer	village_id integer	
1	1101	1201	101
2	1102	1202	102
3	1103	1203	103
4	1104	1204	104
5	1105	1205	105

```

INSERT INTO public."Requirements"("Request_id", requirement_id, request_type)VALUES (1101,
1201, 'IHHL');
INSERT INTO public."Requirements"("Request_id", requirement_id, request_type)VALUES (1102,
1202, 'water connection');
INSERT INTO public."Requirements"("Request_id", requirement_id, request_type)VALUES (1103,
1203, 'water connection');
INSERT INTO public."Requirements"("Request_id", requirement_id, request_type)VALUES (1104,
1204, 'IHHL');
INSERT INTO public."Requirements"("Request_id", requirement_id, request_type)VALUES (1105,
1205, 'water connection');

```

Data Output	Explain	Messages	Notifications
Request_id [PK] integer	requirement_id integer	request_type character varying	
1	1101	1201	IHHL
2	1102	1202	water connection
3	1103	1203	water connection
4	1104	1204	IHHL
5	1105	1205	water connection

```

INSERT INTO public."User_requests"("Request_id", requirement_id, user_id, "Admin_id", "Emp_id",
requirement_id)VALUES (1101, 1201, 1301, 1401, 1501, 'IHHL');
INSERT INTO public."User_requests"("Request_id", requirement_id, user_id, "Admin_id", "Emp_id",
requirement_id)VALUES (1102, 1202, 1302, 1402, 1502, 'water connection');
INSERT INTO public."User_requests"("Request_id", requirement_id, user_id, "Admin_id", "Emp_id",
requirement_id)VALUES (1103, 1203, 1303, 1403, 1503, 'IHHL');
INSERT INTO public."User_requests"("Request_id", requirement_id, user_id, "Admin_id", "Emp_id",
requirement_id)VALUES (1104, 1204, 1304, 1404, 1504, 'water connection');

```

```
INSERT INTO public."User_requests"("Request_id", requirement_id, user_id, "Admin_id", "Emp_id",
requiremnet)VALUES (1105, 1205, 1305, 1405, 1505, 'IHHL');
```






Data Output		Explain	Messages	Notifications		
 Request_id [PK] integer	 requirement_id [PK] integer	 user_id [PK] integer	 Admin_id [PK] integer	 Emp_id [PK] integer	 requiremnet character varying	
1	1101	1201	1301	1401	1501	IHHL
2	1102	1202	1302	1402	1502	water connection
3	1103	1203	1303	1403	1503	IHHL
4	1104	1204	1304	1404	1504	water connection
5	1105	1205	1305	1405	1505	IHHL

INSERT

```
INTO public."Users"(user_id, user_pwd, "Fname", "Lname", "Location_address", village_id)
VALUES (1301, 'ghijklmn', 'monishver', 'sharma', 'chennai', 101);
INSERT INTO public."Users"(user_id, user_pwd, "Fname", "Lname", "Location_address", village_id)
VALUES (1302, '12345678', 'vinay', 'mandru', 'eluru', 102);
INSERT INTO public."Users"(user_id, user_pwd, "Fname", "Lname", "Location_address", village_id)
VALUES (1303, 'abcdefgh', 'abhinay', 'shanmuk', 'ongole', 103);
INSERT INTO public."Users"(user_id, user_pwd, "Fname", "Lname", "Location_address", village_id)
VALUES (1304, '87654321', 'akhil', 'maddu', 'jangareddygudem', 104);
INSERT INTO public."Users"(user_id, user_pwd, "Fname", "Lname", "Location_address", village_id)
VALUES (1304, 'asdfghhjk', 'vineesh', 'reddy', 'tirupathi', 105);
```

Data Output							Explain	Messages	Notifications
	 user_id [PK] integer	 user_pwd character varying (50)	 Fname character varying (50)	 Lname character varying (50)	 Location_address character varying (50)	 village_id integer			
1	1301	ghijklmn	monishver	sharma	chennai	101			
2	1302	12345678	vinay	mandru	eluru	102			
3	1303	abcdefgh	abhinay	shanmuk	ongole	103			
4	1304	87654321	akhil	maddu	jangareddygudem	104			
5	1305	asdfghhjk	vineesh	reddy	tirupathi	105			

```
INSERT INTO public."Employee_details"("Emp_id", emp_pwd, emp_type, emp_age)VALUES (1501,
'123456', 'part time', 40);
INSERT INTO public."Employee_details"("Emp_id", emp_pwd, emp_type, emp_age)VALUES (1502,
'abcdegfh', 'full time', 38);
INSERT INTO public."Employee_details"("Emp_id", emp_pwd, emp_type, emp_age)
VALUES (1503, 'zxdfyui', 'part time', 42);
INSERT INTO public."Employee_details"("Emp_id", emp_pwd, emp_type, emp_age)VALUES (1504,
'lkjhgfdd', 'full time', 35);
INSERT INTO public."Employee_details"("Emp_id", emp_pwd, emp_type, emp_age)VALUES (1505,
'87654321', 'full time', 48);
```

Data Output		Explain	Messages	Notifications
 Emp_id [PK] integer 	emp_pwd character varying 	emp_type character varying 	emp_age integer 	
1	1501	123456	part time	40
2	1502	abcdegfh	full time	38
3	1503	zxdfyui	part time	42
4	1504	lkjhgfd	full time	35
5	1505	87654321	full time	48

```

INSERT INTO public."Employee_salary"("Emp_id", emp_type, emp_salary)VALUES (1501, 'part time',
60000);
INSERT INTO public."Employee_salary"("Emp_id", emp_type, emp_salary)VALUES (1502, 'full time',
50000);
INSERT INTO public."Employee_salary"("Emp_id", emp_type, emp_salary)VALUES (1503, 'part time',
40000);
INSERT INTO public."Employee_salary"("Emp_id", emp_type, emp_salary)VALUES (1504, 'full time',
65000);
INSERT INTO public."Employee_salary"("Emp_id", emp_type, emp_salary)VALUES (1505, 'full time',
80000);

```

Data Output		Explain	Messages	Notifications
 Emp_id [PK] integer 	emp_type character varying 	emp_salary integer 		
1	1501	part time	60000	
2	1502	full time	50000	
3	1503	part time	40000	
4	1504	full time	65000	
5	1505	full time	80000	

```

INSERT INTO public."Administrator"("Admin_id", "Request_id", requirement_id,
project_status)VALUES (1401, 1101, 1201, 'not aprooved');
INSERT INTO public."Administrator"("Admin_id", "Request_id", requirement_id, project_status)
VALUES (1402, 1102, 1202, 'aprooved');
INSERT INTO public."Administrator"("Admin_id", "Request_id", requirement_id, project_status)
VALUES (1403, 1103, 1203, 'aprooved');
INSERT INTO public."Administrator"("Admin_id", "Request_id", requirement_id, project_status)
VALUES (1404, 1104, 1204, 'aprooved');
INSERT INTO public."Administrator"("Admin_id", "Request_id", requirement_id, project_status)
VALUES (1405, 1105, 1205, 'not aprooved');

```

Data Output		Explain	Messages	Notifications
	Admin_Id [PK] integer 	Request_Id [PK] integer 	requirement_Id [PK] integer 	project_status character varying 
1	1401	1101	1201	not aprooved
2	1402	1102	1202	aprooved
3	1403	1103	1203	aprooved
4	1404	1104	1204	aprooved
5	1405	1105	1205	not aprooved

INSERT INTO public."Administrator_login"("Admin_id", admin_pwd)VALUES
(1401,'sweetdaughter');

INSERT INTO public."Administrator_login"("Admin_id", admin_pwd)VALUES (1402, 'nevergiveup');

INSERT INTO public."Administrator_login"("Admin_id", admin_pwd)VALUES (1403, 'motherslove');

INSERT INTO public."Administrator_login"("Admin_id", admin_pwd)VALUES (1404, 'ilovecricket');

INSERT

INTOpublic."Administrator_login"("Admin_id",admin_pwd)VALUES(1405,'krishnabhagavan');

Data Output	Explain	Messages	Notificatio
	Admin_id [PK] integer	admin_pwd character varying	
1	1401	sweetdaughter	
2	1402	nevergiveup	
3	1403	motherslove	
4	1404	ilovecricket	
5	1405	krishnabhagavan	

Table creation comments:

Postgresql is used to create tables in the backend. It has the capability to visualize tables in the UI format. To access that functionality pgadmin will be helpful. From pgadmin all the tables are created in the required fashion

Chapter 8 Frontend design

Introduction to the tool(s):

The tools used in this project for frontend design are HTML5, CSS3, JavaScript, jQuery, Bootstrap.

HTML5 and CSS3 are used in getting the basic layouts with a good design but the pages won't be very much adaptable with all sizes of screens. So, Introduced with bootstrap and now the pages are very much adaptable with all sizes of screens. Then the pages are not reactive, So worked with JavaScript to make the pages responsive. The Graphics are coded using jQuery.

RURAL WATER SUPPLY AND SANITATION

[Home](#)[About Us](#)[Features](#)[Team](#)[Gallery](#)[Login/Register](#)[Contact Us](#)

Employee Registration

[Home](#) / [Employee Registration](#)

First Name

Your name..

Last Name

Your last name..

Employee ID

Your Email Address..

Password

Password..

Salary

Employee Salary

Employee Type

Employee type

Village ID

Your Village name..

Address

Write Address..

Register

Department of CSE, Amrita School of Engineering , Coimbatore

2020Page 63

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Employee Registration

Home / Employee Registration

First Name

Your name..

Last Name

Your last name..

Please fill out this field.

Employee ID

Your Email Address..

Password

Password..

Salary

Employee Salary

Employee Type

Employee type

Village ID

Your Village name..

Address

Write Address..

Register

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Filtration unit registration

Home / filtrationReg

Filtration ID

Filtration ID

Location

Location

Capacity

Capacity

Method

Filtration Method

Submit

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error

Home
About
Privacy Policy
Terms of Use

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Quality Assurance
Home / Quality Assurance

Filtration ID

Chloride

Fluoride

Arsenic

Location

Capacity

Filtration Method

Grade of Contamination:

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error

Home
About
Privacy Policy
Terms of Use

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Project Sanction
Home / project Sanction

Sanction ID

Sanctioned Amount

Year of Sanction

Year of Completion

Deposit Works ID

Max transportation of Water

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error

Home
About
Privacy Policy
Terms of Use

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Village Registration
Home / Villagereg

Village ID
Location

Village Name
Capacity

District Name
Filtration Method

Submit

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error
Home
About
Privacy Policy
Terms of Use

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Water Pipeline Connection
Home / Water Pipeline Connection

Connection ID
Connection ID

Connection type
Connection Type

Length of Pipeline
Length of Pipeline

Year of Laying
Year

Pipe Type
Pipe Type

Flat Rate
Flat Rate

Submit

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error
Home
About
Privacy Policy
Terms of Use

RURAL WATER SUPPLY AND SANITATION

[Home](#)[About Us](#)[Features](#)[Team](#)[Gallery](#)[Login/Register](#)[Contact Us](#)

Water Source Details

Home / Water Source Details

Source ID

Source ID

Source Name

Source Name

Source Capacity

Source Capacity

Maintenance Review Start Date

Maintenance Review Start Date

Maintenance Review End Date

Maintenance Review End Date

Submit

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error

[Home](#)[About](#)[Privacy Policy](#)[Terms of Use](#)

RURAL WATER SUPPLY AND SANITATION

[Home](#)[About Us](#)[Features](#)[Team](#)[Gallery](#)[Login/Register](#)[Contact Us](#)

Login

Home / Login Page

User ID

pottiabhinay2001@gmail.com

Password

Password..

Please fill out this field.

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error

[Home](#)[About](#)[Privacy Policy](#)[Terms of Use](#)

Release a Fund

IHHL Fund ID	Sanctioned IHHL
<input type="text" value="Firstname"/>	<input type="text" value="Lastname"/>
Estimated Cost	
<input type="text" value="1234 Main St"/>	
Funds from NGO	
<input type="text" value="Request ID"/>	
IHHL Id	
<input type="text" value="IHHL ID"/>	
Village ID	
<input type="text" value="Village ID"/>	
<input type="checkbox"/> Agree Terms & Conditions	
<input type="button" value="Submit"/>	

Funds Details

Get Funds				
#	IHHL Fund ID	Sanctioned IHHL	Estimated Cost	Fund From NGO
1	901	5	20000	20000
2	902	7	25000	22000
3	903	5	30000	25000

Request A New IHHL

Firstname	Lastname
<input type="text" value="Firstname"/>	<input type="text" value="Lastname"/>
Address	
<input type="text" value="1234 Main St"/>	
Request ID	
<input type="text" value="Request ID"/>	
Requirement ID	
<input type="text" value="Requirement ID"/>	
Village ID	<input type="checkbox"/> Agree Terms & Conditions
<input type="text" value=""/>	
<input type="button" value="Submit"/>	

Request A New Water Connection

Firstname	Lastname
<input type="text" value="Firstname"/>	<input type="text" value="Lastname"/>
Address	
<input type="text" value="1234 Main St"/>	
Request ID	
<input type="text" value="Request ID"/>	
Requirement ID	
<input type="text" value="Requirement ID"/>	
Village ID	<input type="checkbox"/> Agree Terms & Conditions
<input type="text" value=""/>	
<input type="button" value="Submit"/>	

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Water revenue output
Home / water revenue output

Get Funds

#	Water Revenue Bill ID	Generated Fund
1	401	10000
2	402	20000

© Copyright Syntax_Error. All Rights Reserved
Designed by Syntax_Error

Home About Privacy Policy Terms of Use

RURAL WATER SUPPLY AND SANITATION
Home
About Us
Features
Team
Gallery
Login/Register
Contact Us

Villages
Home / villageoutput

Village ID: 101

Village Name: Podili

District Name: Prakasam

Village ID: 103

Village Name: orugallu

District Name: warangal

Village ID: 102

Village Name: Eluru

District Name: west godavari

Village ID: 104

Village Name: kanigiri

District Name: prakasam

Village ID: 105

Village Name: pamuru

District Name: prakasam

Village ID: 106

Village Name: vadiamudi

District Name: guntur

Village ID: 107

Village Name: kavali

District Name: nellore

Village ID: 108

Village Name: putthur

District Name: chittoor

Village ID: 109

Village Name: srisailem

District Name: kurnool

Village ID: 110

Village Name: Jangareddy gudem

District Name: West godavari

Chapter 9 Database connectivity

Introduction to the connectivity standard:

The Connection with UI and Database was done using the framework Django and the database adapter psycopg2. ODBC standards for python are used to connect with the postgresql. Django uses python language and helps in creating dynamic pages and this project is having many dynamic pages.

Connectivity code:

```
def insertvillage(request):
    try:
        connection = psycopg2.connect(user="postgres",
                                       password="1234",
                                       host="localhost",
                                       database="ruralwatersupplydb")

        cursor = connection.cursor()

        postgres_insert_query = """ INSERT INTO public."Village_Details"("Village_id", "Village_name",
        "District_name") VALUES (%s, %s, %s);"""

        val1 = int(request.GET['villageid'])
        val2 = request.GET['villagename']
        val3 = request.GET['districtname']
        record_to_insert = (val1, val2, val3)
        cursor.execute(postgres_insert_query, record_to_insert)

        connection.commit()
        count = cursor.rowcount
        print (count, "Record inserted successfully into mobile table")

    except (Exception, psycopg2.Error) as error :
        if(connection):
            print("Failed to insert record into mobile table", error)

    finally:
        #closing database connection.
```

```

if(connection):
    cursor.close()
    connection.close()
    print("PostgreSQL connection is closed")

```

```

return render(request,'adminportal.html')

```

```

def insertsanction(request):

```

```

    try:

```

```

        connection = psycopg2.connect(user="postgres",
                                         password="1234",
                                         host="localhost",
                                         database="ruralwatersupplydb")

```

```

        cursor = connection.cursor()

```

```

        postgres_insert_query = """INSERT INTO public."Sanction"("Sanction_id", "Sanctioned_amount",
"Year_of_sanction", "Completion_year") VALUES (%s, %s, %s, %s);"""

```

```

        val1 = int(request.GET['sanctionid'])

```

```

        val2 = int(request.GET['amount'])

```

```

        val3 = int(request.GET['year1'])

```

```

        val4 = int(request.GET['year2'])

```

```

        record_to_insert = (val1, val2, val3, val4)

```

```

        cursor.execute(postgres_insert_query, record_to_insert)

```

```

        val5 = int(request.GET['worksid'])

```

```

        val6 = int(request.GET['maxtranspo'])

```



```

        postgres_insert_query = """ INSERT INTO public."Works"("Works_id",
"Max_Transportation_of_water") VALUES (%s, %s); """
        record_to_insert = (val5, val6)
        cursor.execute(postgres_insert_query, record_to_insert)

        postgres_insert_query = """ INSERT INTO public."Deposit_Works"( "Sanction_id", deposit_works)
VALUES (%s, %s); """
        record_to_insert = (val1, val5)
        cursor.execute(postgres_insert_query, record_to_insert)


        connection.commit()
        count = cursor.rowcount
        print (count, "Record inserted successfully into mobile table")


except (Exception, psycopg2.Error) as error :
    if(connection):
        print("Failed to insert record into mobile table", error)


finally:
    #closing database connection.
    if(connection):
        cursor.close()
        connection.close()
        print("PostgreSQL connection is closed")


return render(request,'adminportal.html')

```

```

urlpatterns = [
    path("", views.index, name='index'),
    path('index', views.index, name='index'),
    path('register', views.register, name='register'),
    path('fundinputpage', views.fundinputpage, name='fundinputpage'),
    path('login', views.login, name='login'),
    path('adminportal', views.adminportal, name='adminportal'),
    path('empreg', views.empreg, name='empreg'),
    path('filtrationunit', views.filtrationunit, name='filtrationunit'),
    path('quality', views.quality, name='quality'),
    path('sanction', views.sanction, name='sanction'),
    path('villagereg', views.villagereg, name='villagereg'),
    path('pipeline', views.pipeline, name='pipeline'),
    path('watersourcereg', views.watersourcereg, name='watersourcereg'),
    path('insertvillage', views.insertvillage, name='insertvillage'),
    path('empinsert', views.employeeereg, name='employeeereg'),
    path('insertfiltration', views.insertfiltration, name='insertfiltration'),
    path('insertquality', views.insertquality, name='insertquality'),
    path('insertsanction', views.insertsanction, name='insertsanction'),
    path('insertpipe', views.insertpipe, name='insertpipe'),
    path('insertsource', views.insertsource, name='insertsource'),
    path('userreg', views.userreg, name='userreg'),
    path('userlogin', views.userlogin, name='userlogin'),
    path('reqwatercon', views.reqwatercon, name='reqwatercon'),
    path('reqihhl', views.reqihhl, name='reqihhl'),
    path('funding', views.fundretrieve, name='funding'),
    path('newwaterconreq', views.newwaterconreq, name='newwaterconreq'),
    path('newihhlcon', views.newihhlcon, name='newihhlcon'),
    path('fundretrieve', views.fundretrieve, name='fundretrieve'),
    path('insertfund', views.insertfund, name='insertfund'),
    path('waterrevenueinput', views.waterrevenueinput, name='waterrevenueinput'),
    path('waterrevenueretrieve', views.waterrevenueretrieve, name='waterrevenueretrieve'),
    path('insertwaterrevenue', views.insertwaterrevenue, name='insertwaterrevenue'),

```

```

    path('registerihhl',views.registerihhl,name='registerihhl'),
    path('insertihhl',views.insertihhl,name='insertihhl'),
    path('villageoutput',views.villageoutput,name='villageoutput')

]

```

Chapter 10 Sample code

Sample code of selected UI screens

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Fund Input</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

    </head>
<body>

<nav class="navbar navbar-inverse">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">Rural Water Supply and Sanitation</a>
        </div>
        <ul class="nav navbar-nav">
            <li><a href="userhome.html">Home</a></li>
            <li><a href="reqwatercon.html">Request New Water Connection</a></li>
            <li><a href="reqihhl.html">Request New IHHL</a></li>
            <li><a href="funding.html">Funding Details</a></li>
            <li class="active"><a href="fundinput.html">Release Fund</a></li>

```

```

    <li ><a href="index.html" onclick="logout()">Logout</a></li>
</ul>

</div>
</nav>
<h1>Release a Fund</h1>

<div class="container">
    <form onsubmit="myFunction()" action="insertfund" >
<div class="form-row">
    <div class="form-group col-md-6">
        <label for="inputName">IHHL Fund ID</label>
        <input type="name" class="form-control" id="inputName" placeholder="Firstname"
name="ihhl_fund_id" required>
    </div>
    <div class="form-group col-md-6">
        <label for="inputLastname">Sanctioned IHHL</label>
        <input type="name" class="form-control" id="inputLastname" placeholder="Lastname"
name="sanctioned_ihhl" required>
    </div>
</div>
<div class="form-group">
    <label for="inputAddress">Estimated Cost</label>
    <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St"
name="estimated_cost" required>
</div>
<div class="form-group">
    <label for="request">Funds from NGO</label>
    <input type="text" class="form-control" id="request" placeholder="Request ID"
name="funds_from_ngo" required>
</div>

```

```

<div class="form-group">
  <label for="request">IHHL Id</label>
  <input type="text" class="form-control" id="request" placeholder="IHHL ID" name="ihhlid"
required>
</div>

<div class="form-group">
  <label for="request">Village ID</label>
  <input type="text" class="form-control" id="request" placeholder="Village ID" name="villageid"
required>
</div>

<div class="form-group">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="gridCheck" required>
    <label class="form-check-label" for="gridCheck">
      Agree Terms & Conditions
    </label>
  </div>
</div>

<!-- <div class="form-group"> -->
  <!-- <label for="requirement">Requirement ID</label> -->
  <!-- <input type="text" class="form-control" id="requirement" placeholder="Requirement ID"
name="requirementid" required> -->
<!-- </div> -->

<!-- <div class="form-row"> -->
  <!-- <div class="form-group col-md-6"> -->
    <!-- <label for="inputVillage">Village ID</label> -->
    <!-- <input type="text" class="form-control" id="inputVillage" name="villageid" required> -->
  <!-- </div> -->
  <!-- <div class="form-group col-md-4"> -->
    <!-- <label for="inputState">State</label> -->
    <!-- <select id="inputState" class="form-control" required> -->
      <!-- <option selected>Choose...</option> -->
      <!-- <option>...</option> -->

```

```

        <!-- </select> -->
    <!-- </div> -->
    <!-- <div class="form-group col-md-2"> -->
        <!-- <label for="inputZip">Zip</label> -->
        <!-- <input type="number" class="form-control" id="inputZip" required> -->
    <!-- </div> -->
    <button type="submit" class="btn btn-primary" id="connsubmit">Submit</button>
</div>

    <!-- <div class="form-group col-md-12"> -->
        <!-- <label for="inputconnection">Connection Type</label> -->
        <!-- <select id="inputconnection" class="form-control" required> -->
            <!-- <option selected>Choose...</option> -->
            <!-- <option></option> -->
        <!-- </select> -->
    <!-- </div> -->
    <br>

    <br>

</form>
<!--      <input type="submit" class="btn btn-primary" id="connsubmit">-->
    <p id="success"></p>

</div>
<script>

//  document.getElementById("connsubmit").onclick = function(){
//      document.getElementById("success").innerHTML = "Request Submitted";
//
//  }
</script>
<script>
function logout(){

```

```

        alert("Logged Out Successfully");
    }
    function myFunction(){
        alert("Request Submitted");
    }

</script>
</body>
</html>

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Funded Details</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>
<body>

<nav class="navbar navbar-inverse">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">Rural Water Supply and Sanitation</a>
        </div>
        <ul class="nav navbar-nav">
            <li><a href="userhome.html">Home</a></li>
            <li><a href="reqwatercon.html">Request New Water Connection</a></li>

```

```

<li><a href="reqihhl.html">Request New IHHL</a></li>
<li class="active"><a href="funding.html">Funding Details</a></li>
<li><a href="fundinput.html">Release Fund</a></li>
<li><a href="index.html" onclick="logout()">Logout</a></li>
</ul>

```

```

</div>

```

```

</nav>

```

```

<h1 align="center">Funds Details</h1>
<div class="container">
  <form action="fundretrieve">
    <input type="submit" value="Get Funds">

```

```

  </form>

```

```

</div>

```

```

<div class="container">

```

```

  <table class="table table-bordered">

```

```

<thead>

```

```

  <tr>

```

```

    <th scope="col">#</th>

```

```

    <th scope="col">IHHL Fund ID</th>

```

```

    <th scope="col">Sanctioned IHHL</th>

```

```

    <th scope="col">Estimated Cost</th>

```

```

    <th scope="col">Fund From NGO</th>

```

```

  </tr>

```

```

</thead>

```

```

<tbody>

```

```

  {% for fund in funddetails %}

```



```

<tr>
  <th scope="row">{{ fund.s }}</th>

  <td>{{ fund.IHHL_fund_id }}</td>
  <td>{{ fund.Sanctioned_IHHL }}</td>
  <td>{{ fund.Estimated_Cost }}</td>
  <td>{{ fund.Funds_from_ngo }}</td>
</tr>
{% endfor %}

</tbody>
</table>

</div>
<script>
  function logout(){
    alert("Logged Out Successfully");
  }

</script>
</body>
</html>

```

Chapter 11 Conclusions

Elucidating important features of the project

- Our project provides complete end to end tools and applications for rural water supply and sanitation management. We have included pages for new signups, logins separately for users and admins. This has been made to avoid unwanted details shown to normal customers and also administrators have to take actions regarding the ongoing projects and pass them on to employees. User requests for rural water supply and IHHL can be made by the users and passed on to employees and admins to process it. We keep track of the details of various non-government organizations, and to display funds required for upcoming projects. Also, we have created pages where the NGOs can help by providing funds for the projects, and this will be updated dynamically. We can also visualize the funds allotted for villages with the projects done.

Chapter 12 References

Books references:

Database Management Systems by Ramakrishnan and Gehrke, McGraw Hill

Websites references: