



REVIEW : 2

Project Name:

CLEAN INDIA MISSION

BY

22C46039 – Tanshi Gandhi

**BRANCH : Bachelor's Of Computer
Applications**

Semester – 6th

**Submitted to Internal Guide:
Prof. Ankit Vyas**

Abstract

The **Clean India Mission**, also known as **Swachh Bharat Abhiyan**, is an ambitious initiative launched by the Government of India with the goal of cleaning the streets, roads, and infrastructure of Indian cities, villages, and towns. This mission aims to promote cleanliness, hygiene, and sanitation across the nation, ultimately leading to improved public health, environmental sustainability, and a higher standard of living for citizens.

This project, *Clean India Mission*, leverages technology to enhance the impact and outreach of the mission. The primary objective of the project is to provide an efficient platform that enables better tracking of cleanliness activities, promotes awareness about sanitation, and allows citizens to report issues related to waste disposal, littering, and sanitation.

The project utilizes a **web-based application** or **mobile app** that integrates features such as geolocation for waste tracking, public awareness campaigns, a reporting system for unclean areas, and a feedback mechanism to evaluate the effectiveness of cleanliness initiatives. Through real-time data collection, citizens, municipal bodies, and local authorities can collaborate to improve sanitation practices.

By combining technology with social responsibility, the *Clean India Mission* aims to achieve the broader objective of a cleaner, healthier, and more sustainable India, ensuring that every individual is encouraged to take responsibility for their environment.

This project is aimed at **empowering citizens** to actively participate in cleanliness efforts while providing municipal bodies with the tools they need to monitor and improve sanitation across urban and rural areas. In the long term, it will help in creating cleaner and healthier environments, reducing pollution, and contributing towards India's sustainable development goals.

Introduction

The **Clean India Mission** (Swachh Bharat Abhiyan) is a nationwide campaign initiated by the Government of India on October 2, 2014, with the vision of achieving a cleaner, healthier, and more hygienic India. The mission focuses on improving sanitation standards, eliminating open defecation, promoting waste management, and creating public awareness around the importance of cleanliness. By targeting both urban and rural areas, it seeks to create a paradigm shift in the way individuals perceive cleanliness and hygiene.

This project, **Clean India Mission**, is aligned with the broader objectives of the government initiative, but with a focus on utilizing **technology** to accelerate and support its goals. The project aims to create a comprehensive platform that empowers both citizens and local authorities to actively engage in maintaining cleanliness and improving sanitation infrastructure.

The digital platform developed in this project serves several critical purposes:

1. **Tracking Cleanliness Efforts:** It provides a mechanism to track the progress of cleanliness campaigns at various locations, enabling real-time monitoring and assessment of sanitation conditions.
2. **Public Engagement:** It encourages active participation from citizens by allowing them to report issues such as waste accumulation, improper disposal of garbage, and the presence of littered areas in their communities.

3. **Feedback Mechanism:** Citizens can provide feedback regarding cleanliness initiatives, helping local authorities gauge the effectiveness of implemented policies and measures.
4. **Awareness Programs:** The platform includes resources that educate users about the importance of cleanliness, hygiene, and sustainable waste management practices.

Through this project, the aim is to facilitate better collaboration between citizens, government bodies, and local authorities, ultimately working toward creating an environment that is cleaner, healthier, and more sustainable for future generations.

The **Clean India Mission** is not just a one-time effort but a long-term vision for a cleaner India, requiring collective responsibility, behavior change, and consistent efforts across every segment of society. By integrating technology into the mission, we envision a more efficient, transparent, and impactful approach to cleanliness, which will contribute to the country's overall development and well-being.

1. Project Task Planning

Project Planning

- We Finalize project scope and objectives
- Identify key stakeholders and establish communication plan
- Develop project timeline and schedule
- Create project team roles and responsibilities
- Establish project management tools and processes

Requirements Gathering

- Identify and document requirements for the Preschool management system
- Conduct Communication with stakeholders to gather

- Create use cases and scenarios for the system
- Create a functional requirements document

System Design

- Create system architecture and design
- Identify required technologies and tools
- Understand database schema and data model
- Create wireframes and user interface design
- Understand system specifications and requirements

Learning web Development Phase

- Learn HTML
- Learn CSS
- Learn Bootstrap
- Start user interface design

2. Data Dictionary

Definition :

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects

relationships to other objects, and other data. The data dictionary is a crucial component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

Table 1: Admin_mst

Description: - Information about admin

Sr.	Field Name	Type	Constraint	Description
1.	Admin_id	Number	Primary Key	Admin ID
2.	email	String	Not Null	Email ID
3.	pwd	String	Not Null	Password

Table 2: Donor_mst

Description: - Information About Donor

Sr.	Field Name	Type	Constraint	Description
1.	Donor_id	Number	Primary Key	Donor ID
2.	email	String	Foreign Key	Email ID
3.	pwd	String	Not Null	Password

Table 3: CMS Block_mst

Description: - Information about cms block

Sr.	Field Name	Type	Constraint	Description
1.	cms_id	Number	Primary Key	Cms block ID
2.	title	String	Not Null	Cms Block Title
3.	identifier	String	Null	Cms block identifier
4.	dsc	String	Not Null	Description of cms block
5.	img	String	Not Null	Image of block
6.	status	String	Not Null	Status of cms block
7.	created_At	Datetime	Not Null	Creation time of block
8.	updated_At	Datetime	Not Null	Updation time of block

Table 4: ContactUs_mst

Description: - Information about contact us page

Sr.	Field Name	Type	Constraint	Description
1.	id	Number	Primary Key	Contact us page ID
2.	name	String	Not Null	Name
3.	email	String	Not Null	Email
4.	Sbj	String	Not Null	Subject
5.	msg	String	Not Null	Message
6.	created_At	Datetime	Not Null	Creation Time
7.	updated_At	Datetime	Not Null	Updaation Time

Table 5: State_mst

Description: - Information About State

Sr.	Field Name	Type	Constraint	Description
1.	id	Number	Primary Key	State ID
2.	state_name	String	Foreign Key	State Name
3.	state_code	String	Not Null	State Code
4	created_at	Datetime	Not Null	Creation Time
5.	updated_at	Datetime	Not Null	Updation Time

Table 6: Donation_mst

Description: - Information About Donation

Sr.	Field Name	Type	Constraint	Description
1.	ID	Number	Primary Key	Donation ID
2.	amt	Number	Not Null	Amount
3.	name	String	Not Null	Name
4.	email	String	Foreign Key	Email ID
5.	phn	Number	Not Null	Phone Number
6.	pan_no	String	Not Null	PAN Number
7.	address	String	Not Null	Address
8.	State_name	String	Foreign Key	State Name
9.	city	String	Not Null	City Name
10.	trsn_id	String	Null	Transaction Id

Table 7: Language_mst

Description: - Information About Languages

Sr.	Field Name	Type	Constraint	Description
1.	id	Number	Primary Key	Language ID
2.	lang_code	String	Not Null	Language Code
3.	lang_name	String	Not Null	Language Name
4.	status	String	Not Null	Status

Table 8: Volunteer_mst

Description: - Information About Volunteer

Sr.	Field Name	Type	Constraint	Description
1.	id	Number	Primary Key	Volunteer's ID
2.	vol_name	String	Foreign Key	Volunteer Name
3.	email	String	Not Null	Email
4.	state_name	String	Foreign key	State Name
5.	city	String	Not Null	City
6.	phn	Number	Not Null	Phone Number
7.	img	String	Null	Image

Table 9: Program_mst

Description: - Information About Programs

Sr.	Field Name	Type	Constraint	Description
1.	id	Number	Primary Key	Program ID
2.	prog_place	String	Not Null	Program Place
3.	desc	String	Not Null	Description
4.	why_this_place	String	Not Null	Reason for this place
5.	state_name	String	Foreign Key	State Name
6.	city	String	Not Null	City
7.	vol_name	String	Foreign Key	Volunteer Name
8.	date	Datetime	Not Null	Date
9.	area_cover	Number	Null	Area Cover
10.	status	String	Not null	Status

3. Class Diagram

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

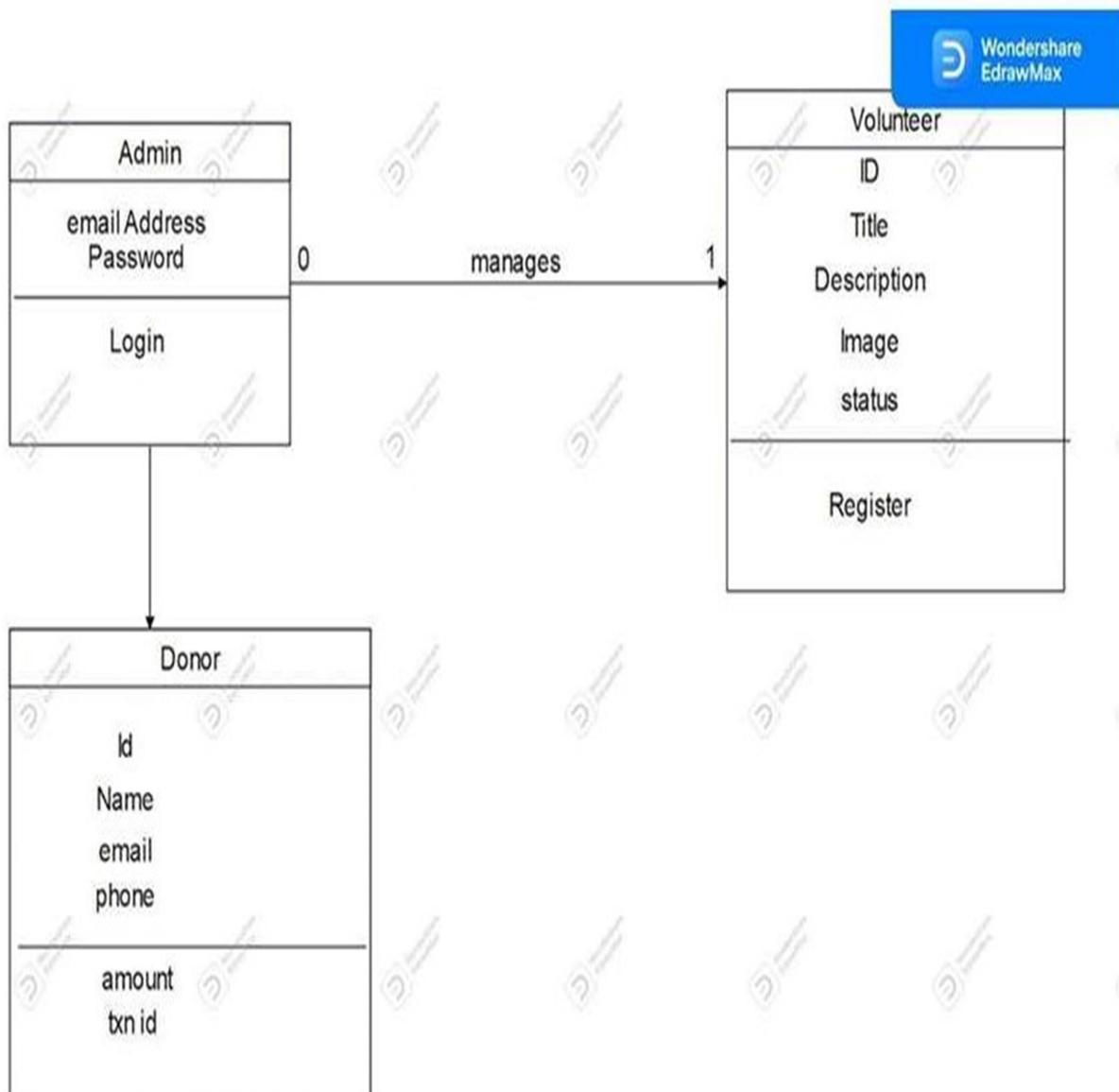
The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling the classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

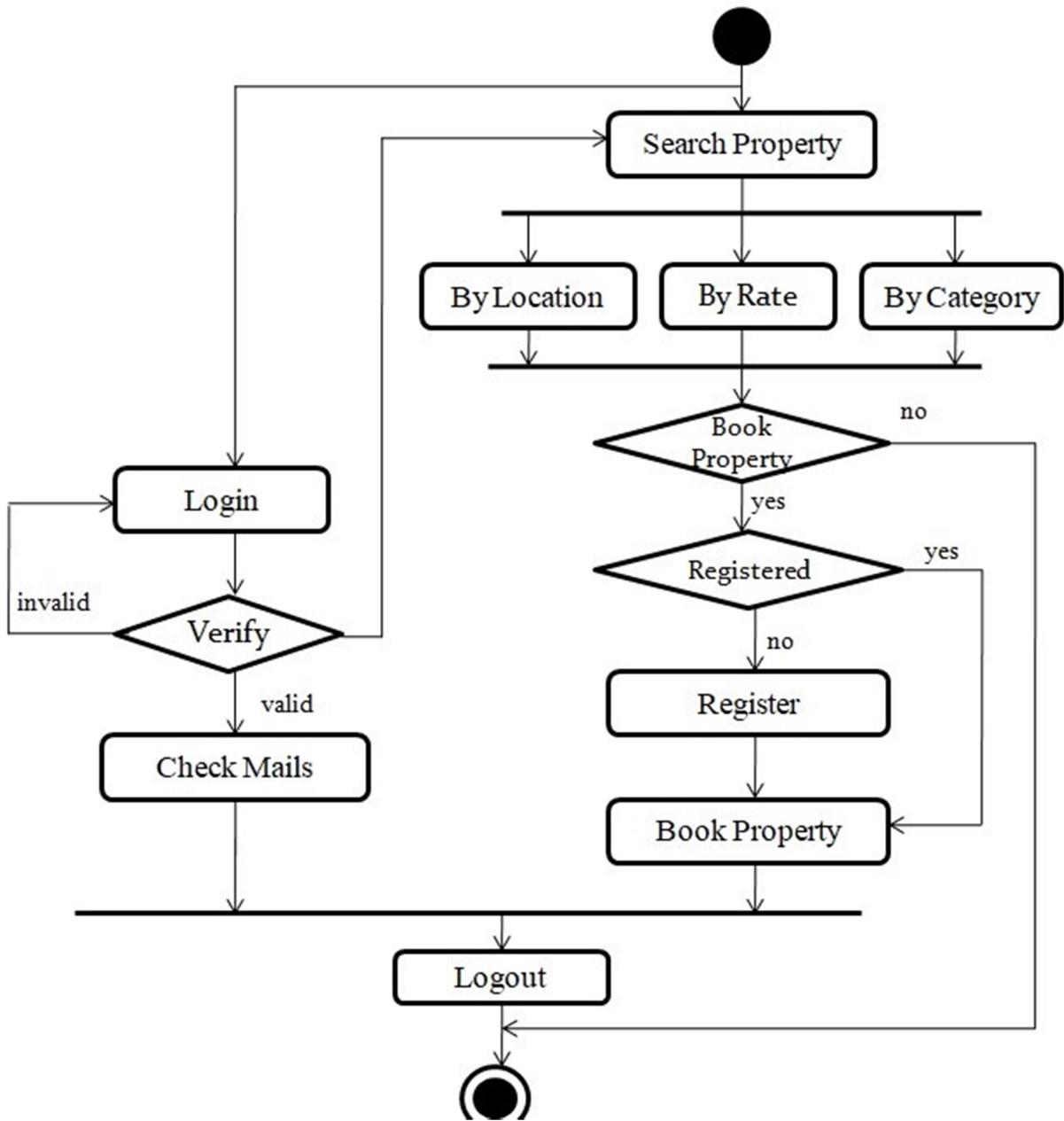
In the diagram, classes are represented with boxes that contain three compartments:

- ❖ The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- ❖ The middle compartment contains the attributes of the class. They are left-aligned, and the first letter is lowercase.

- ❖ The bottom compartment contains the operations the class can execute. They are also left-aligned, and the first letter is lowercase.

In the design of a system, a few classes are identified and grouped together in a class diagram that helps to determine the static relations between them. With detailed modelling, the classes of the conceptual design are often split into a few subclasses.





4. Project screenshots (30% coding)

Coding Standards

1. Admin Login Page

```
40
41 router.get('/login',(req,res,next)=>{
42   User.find({email:req.body.email})
43   .exec()
44   .then(user=>{
45     if(user.length < 1)
46     {
47       return res.status(401).json({
48         msg : 'User not exist'
49       })
50     }
51     bcrypt.compare(req.body.password,user[0].password,(err,result)=>{
52       if(!result)
53       {
54         return res.status(401).json({
55           msg : 'password matching fail'
56         })
57       }
58       if(result)
59       {
60         const token = jwt.sign({
61           role:user[0].role,
62           email:user[0].email
63
64           },
65           "this is dummy text",
66           {
67             expiresIn:"24h"
68           }
69         );
70         res.status(200).json({
71           token:token
72         })
73       }
74     })
75   })
76   .catch(err=>{
77     res.status(500).json({
78       error:err
79     })
80   })
81 })
82 })
83 })
```

2. Donor Signup

```
router.post('/signup',(req,res,next)=>{
  bcrypt.hash(req.body.password,10,(err,hash)=>{
    if(err)
    {
      return res.status(500).json({
        error:err
      })
    }
    else
    {
      const user = new User({
        username:req.body.username,
        password:hash,
        email:req.body.email,
        role:"donor"
      })
      +
      user.save()
      .then(result=>{
        res.status(200).json({
          new_user:result
        })
      })
      .catch(err=>{
        res.status(500).json({
          error:err
        })
      })
    }
  })
})
```

3. Donor Login

```
40
41 router.get('/login',(req,res,next)=>{
42   User.find({email:req.body.email})
43     .exec()
44     .then(user=>{
45       if(user.length < 1)
46       {
47         return res.status(401).json({
48           msg : 'User not exist'
49         })
50       }
51       bcrypt.compare(req.body.password,user[0].password,(err,result)=>{
52         if(!result)
53         {
54           return res.status(401).json({
55             msg : 'password matching fail'
56           })
57         }
58         if(result)
59         {
60           const token = jwt.sign({
61             role:user[0].role,
62             email:user[0].email
63           },
64           "this is dummy text",
65           {
66             expiresIn:"24h"
67           }
68         );
69         res.status(200).json({
70           token:token
71         })
72       }
73     })
74   })
75   .catch(err=>{
76     res.status(500).json({
77       error:err
78     })
79   })
80 })
81 })
82 })
83 })
```

4. Home Page



5. Admin Login Page

A screenshot of an admin login interface. It features a central modal window with a light gray background. Inside the modal, at the top, is the text "Sign in to start your session". Below this are two input fields: one for "E-Mail Address" and one for "Password". Underneath the password field is a reCAPTCHA verification box containing the text "I'm not a robot" next to a checkbox and the reCAPTCHA logo. At the bottom of the modal is a blue "Login" button. The entire login form is set against a light gray background.

6. Admin Update Page

The screenshot shows the 'Edit Admin Detail' page. At the top right are links for 'Dashboard', 'Alok', and 'Logout'. The main form has fields for 'Name:' (Alok), 'E-Mail Address:' (empty), 'Password:' (empty), 'Confirm Password:' (empty), and a dropdown 'Role:' set to 'admin'. Below the form are 'Back' and 'Update' buttons.

Name:*	Alok
E-Mail Address:*	
Password:*	
Confirm Password:*	
Role:*	admin

Back Update

7. Admin Control Panel

The screenshot shows the Admin Control Panel dashboard. On the left is a sidebar with 'Master' and 'Setup' sections. Under 'Master' are links for State, Language, Country, Pages, Volunteer, Slider, OurImpact, Takeapledge, Programmes, Donation, and Registervolunteer. Under 'Setup' are links for Cmsblock, Contacts, and Admin. The main area displays four cards: 'Total Programmes' (8), 'Total Donors' (78), 'Total Take a Pledge' (23), 'Total OurImpact' (6), 'Total Joined Volunteers' (10), and 'Total Apply Volunteers' (34). The top right shows a 'Home' link and a 'Dashboard' link.

Total Programmes	8
Total Donors	78
Total Take a Pledge	23
Total OurImpact	6
Total Joined Volunteers	10
Total Apply Volunteers	34

THANK YOU !!