



A simple web-based email program

Name 1 : Abdelrahman Ashraf Elmenyawy
ID 1 : 18012538

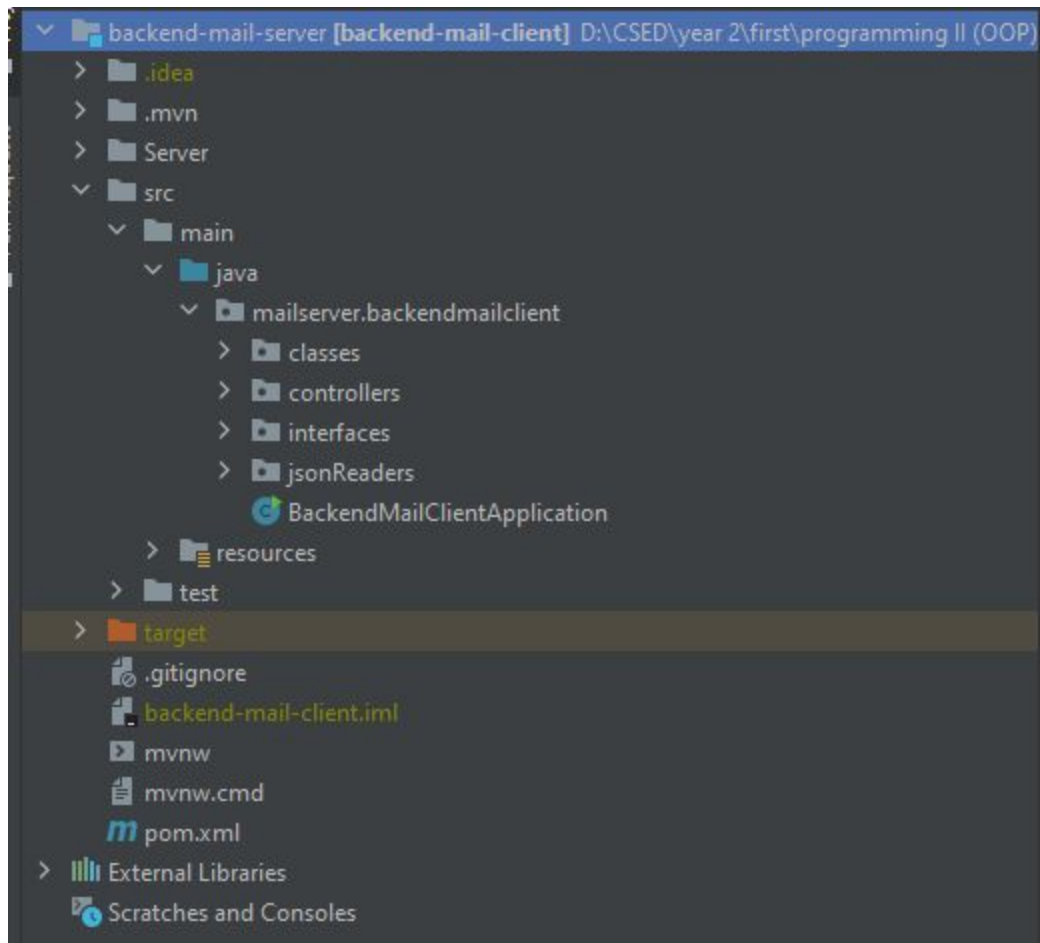
Name 3 : Mark Magdy Nasr
ID 3 : 18011304

Name 2 : Mina Henen Shafik
ID 2 : 18011939

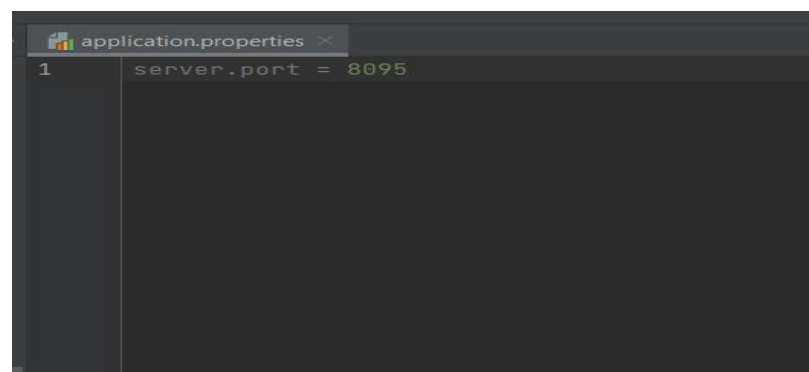
Name 4 : Mark Nader Fathy
ID 4 : 18011305

The steps required to run the code :

- Import backend-paint folder to IDE (like intellij)



- Run the backend
- It will run on server port 8095




- Import frontend-paint to visual studio code
- Write in terminal

Yarn install

Yarn serve

[Login](#) | [Register](#)



Demo Mail

Sign in

E-mail

Password

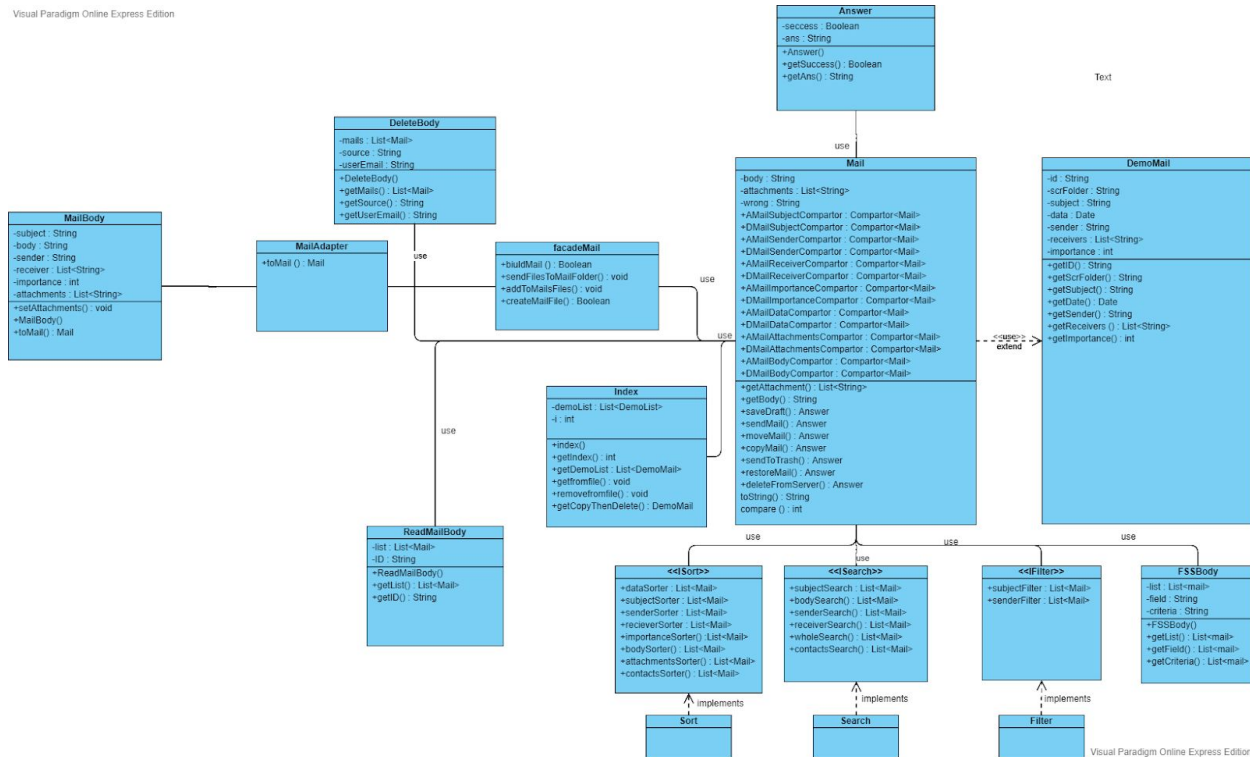
Login

[No account yet?](#)

UML diagram :

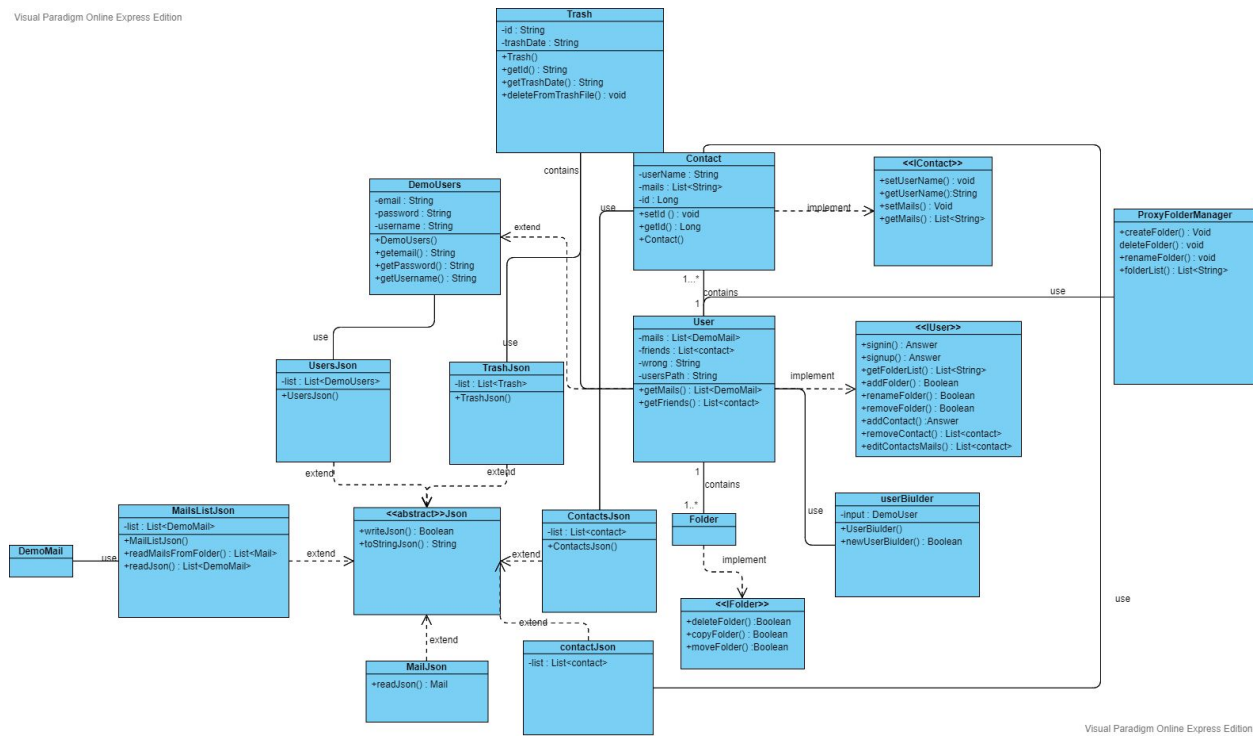
★ Class diagram

Visual Paradigm Online Express Edition

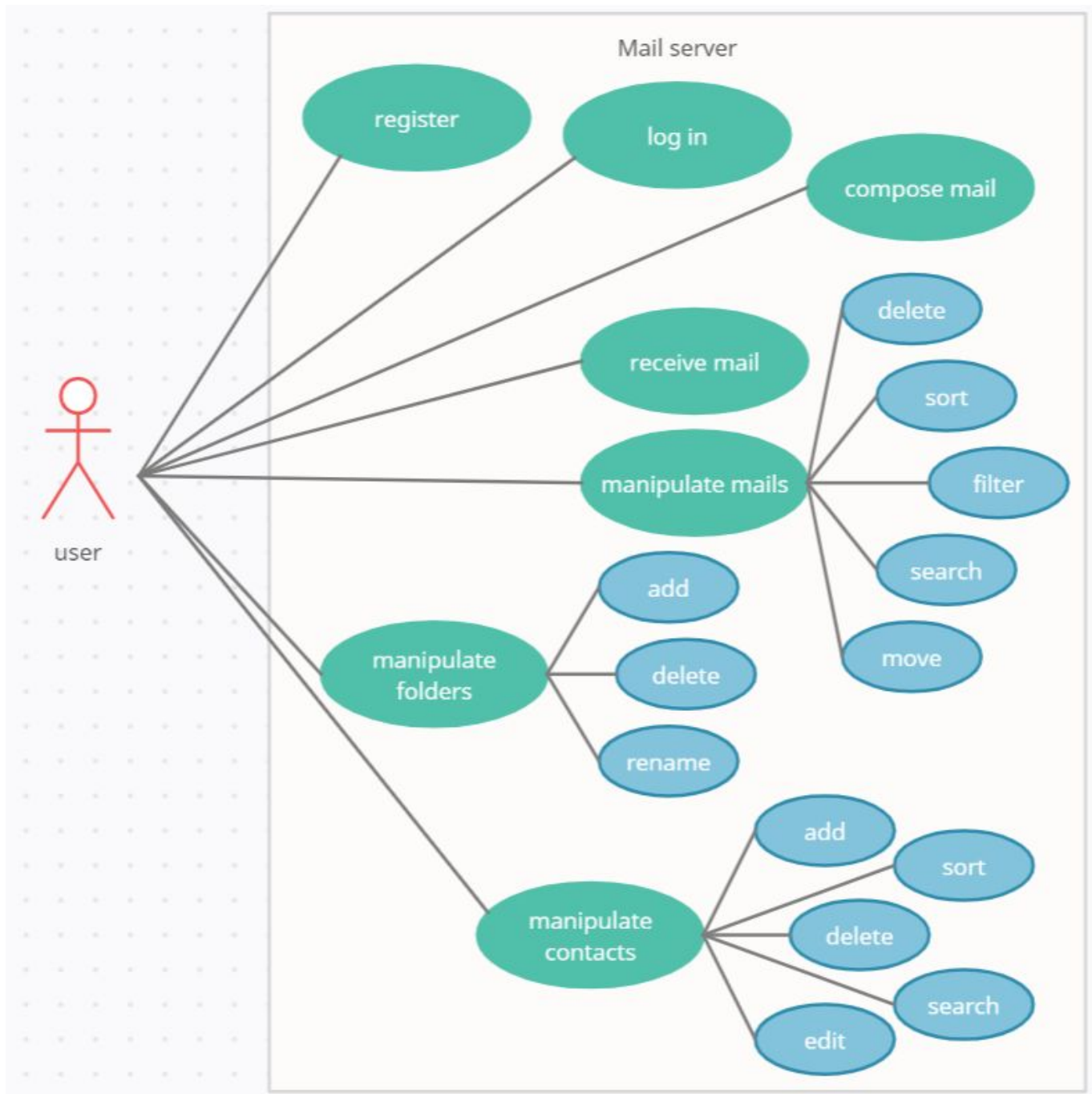


Text

Visual Paradigm Online Express Edition



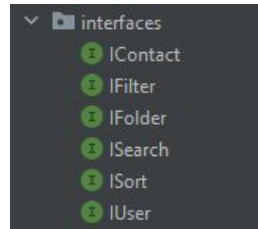
★ Use-case diagram



Design pattern in the code:

➤ Basic design patterns:

- Interface design pattern: (interfaces)



➤ Creational design patterns:

- ★ Singleton design pattern: (singleton mail server)

```
public class MailServer {

    private static MailServer mailserver = null;

    private MailServer() {
    }

    public static MailServer getInstance() {
        if (mailserver == null) {
            mailserver = new MailServer();
        }
        return mailserver;
    }
}
```

- ★ Factory design pattern: (Json reader factory)

```
public class JsonFactory {

    public Json jsfactory(ReaderType t, Object list) {
        switch (t) {
            case MAIL:
                return new MailJson();
            case CONTACTS:
                return new ContactsJson((List<Contact>) list);
            case MAILSLIST:
                return new MailsListJson((List<DemoMail>) list);
            case TRASH:
                return new TrashJson((List<Trash>) list);
            case USERS:
                return new UsersJson((List<DemoUsers>) list);
        }
        return null;
    }
}
```

★ Builder design pattern: (user builder)

```
public class UserBuilder {
    DemoUsers input;

    public UserBuilder(DemoUsers input) { this.input = input; }

    public boolean newUserBuilder() {
        String[] folders = { "Inbox", "Sent", "Draft", "Trash" };
        File userfolder = new File( parent: "Server", input.getemail());
        userfolder.mkdir();
        File contactsFolder = new File(userfolder, child: "contacts.json");
        try {
            contactsFolder.createNewFile();
        } catch (IOException e1) {
            e1.printStackTrace();
            return false;
        }
        userfolder = new File(userfolder, child: "folders");
        userfolder.mkdir();
        File changeableFolders = new File(userfolder, child: "UserFolders");
        changeableFolders.mkdir();

        for (String folder : folders) {
            File currfolder = new File(userfolder, folder);
            currfolder.mkdir();
            File mails = new File(currfolder, child: "mails.json");
            try {
                mails.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
                return false;
            }
            if (folder.compareTo("Trash") == 0) {
                mails = new File(currfolder, child: "Trashfile.json");
                try {
                    mails.createNewFile();
                } catch (IOException e) {
                    e.printStackTrace();
                    return false;
                }
            }
        }
        return true;
    }
}
```


➤ Partitioning design patterns:

- ★ Filter design pattern: (mails filter)

```
public class Filter implements IFilter {

    @Override
    public List<Mail> subjectFilter(List<Mail> mails, String criteria) {
        List<Mail> filteredList = new ArrayList<>();
        for (Mail mail : mails){
            if(mail.getSubject().equalsIgnoreCase(criteria)){
                filteredList.add(mail);
            }
        }
        return filteredList;
    }

    @Override
    public List<Mail> senderFilter(List<Mail> mails, String criteria) {
        List<Mail> filteredList = new ArrayList<>();
        for (Mail mail : mails){
            if(mail.getSender().equalsIgnoreCase(criteria)){
                filteredList.add(mail);
            }
        }
        return filteredList;
    }
}
```

➤ Structural design patterns:

- ★ Proxy design pattern: (proxy folder manager)

```
public class ProxyFolderManager {

    User user = new User();

    public void createFolder(String user, String name) { this.user.addFolder(user, name); }

    public void deleteFolder(String user, String name) { this.user.removeFolder(user, name); }

    public void renameFolder(String user, String oldName, String newName) { this.user.renameFolder(user, oldName, newName); }

    public List<String> foldersList(String username) { return this.user.getFolderList(username); }

}
```

★ Adapter design pattern: (mail adapter)

```
public class mailAdapter {
    public Mail toMail(MailBody mailBody) {
        Mail ans = new Mail(mailBody.getSender(), mailBody.getReceivers(), mailBody.getSubject(), mailBody.getBody(),
            mailBody.getAttachments(), mailBody.getImportance());
        Long time = System.currentTimeMillis();
        String mailname = Long.toString(time);
        ans.setID(mailname);
        ans.setDate(new Date(time));
        return ans;
    }
}
```

★ Façade design pattern: (Façade mail)

```
public class FaçadeMail {
    public boolean buildMail(Mail mail) throws IOException {...}

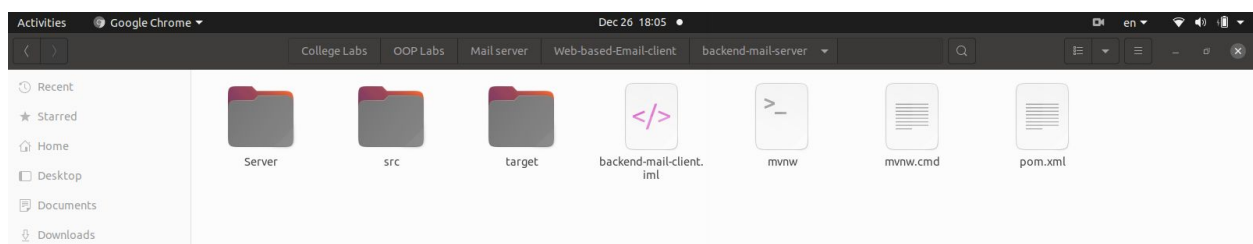
    void sendFilesToMailFolder(Mail mail, File mailfolder, List<MultipartFile> files) throws IOException {...}

    public void addToMailsFile(File mainfolder, DemoMail temp) {...}

    private boolean createMailFile(File mailfolder, Mail mail) {...}
}
```

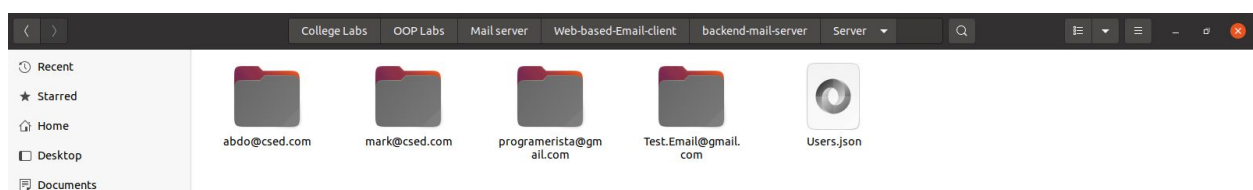
Json Schema:

All the data are saved in a folder called Server sided with the src folder



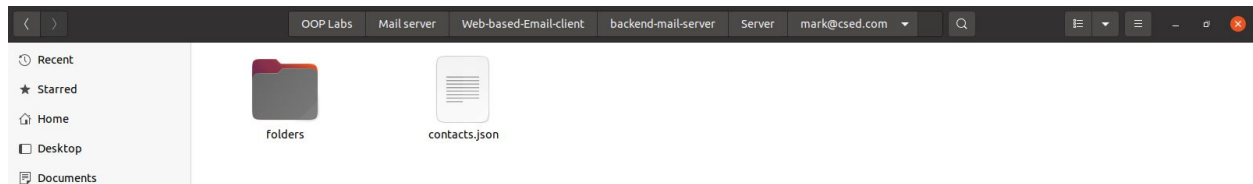
Server Folder contains:

- 1- Users.json : File storing all users data (email addresses, passwords, usernames)
- 2- Folder for each user data



Each user Folder contains:

- 1- Contacts.json : File storing all contacts of this user
- 2- Folder containing all emails data for this user



Now let's open folders:



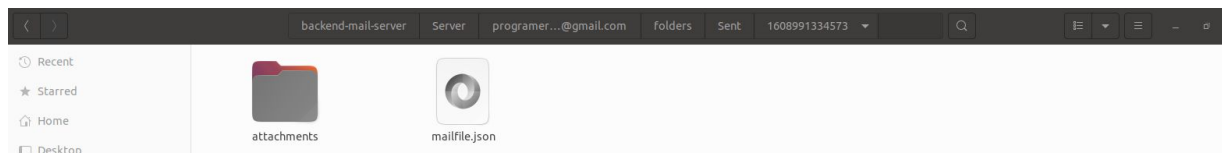
Each category folder contains:

- 1- mails.json : File storing all emails headers (subject, date, sender, receivers)
- 2- Folder for each email containing its specified data

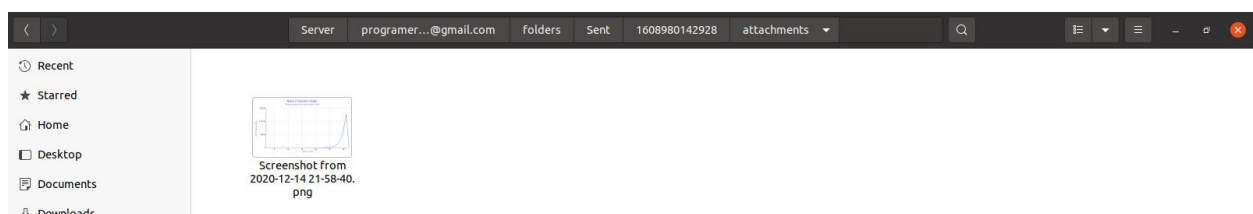


Each email folder contains:

- 1- mailfile.json : File storing all email data
- 2- attachments folder storing all the attached files with the email




Finally we opened the attachments folder:



Screenshots of UI :

- When you run front end

[Login](#) | [Register](#)



Demo Mail

Sign in

E-mail


Password

Login

[No account yet?](#)

- If you don't have email click **No account yet** or **Register**

[Login](#) | [Register](#)



Demo Mail

Sign in

E-mail


Password

Login

[No account yet?](#)

- Fill the data then click register

[Login](#) | [Register](#)



Demo Mail

Welcome

Full Name

E-mail

Password


Confirm

password must contains at least 6 characters, 1 number and 1 uppercase letter

Register

[Already have an account?](#)

[Login](#) | [Register](#)



Demo Mail

Welcome

Full Name

E-mail

Password

Confirm

password must contains at least 6 characters, 1 number and 1 uppercase letter

Register

[Already have an account?](#)

If email is already exist you will get:

localhost:8081 says
This email already exists

OK

Demo Mail

Welcome

Full Name

test

E-mail

test@demo.com

Password

Confirm

*****|

password must contains at least 6 characters, 1 number and 1 uppercase letter

Register

[Already have an account?](#)

If password is invalid:

Full Name

test

E-mail

test@demo.com

Password

Confirm

|

password must contains at least 6 characters, 1 number and 1 uppercase letter

Register

Password does not contain uppercase letter

Full Name

test

E-mail

test@demo.com

Password

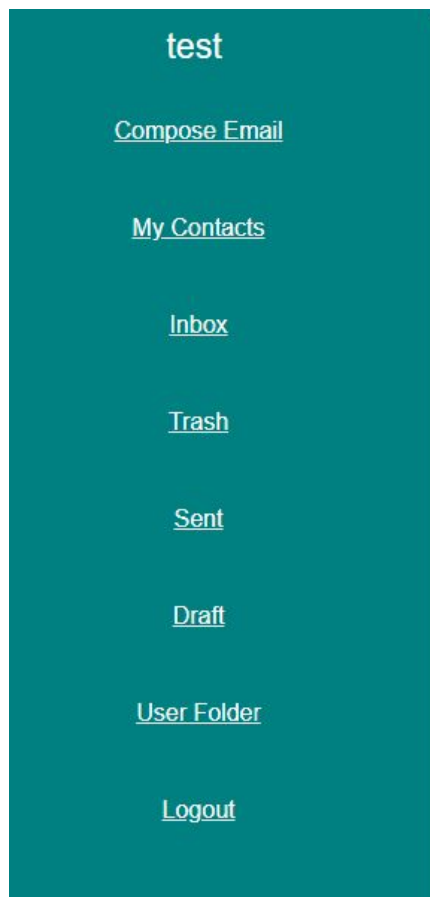
Confirm

password must contains at least 6 characters, 1 number and 1 uppercase letter

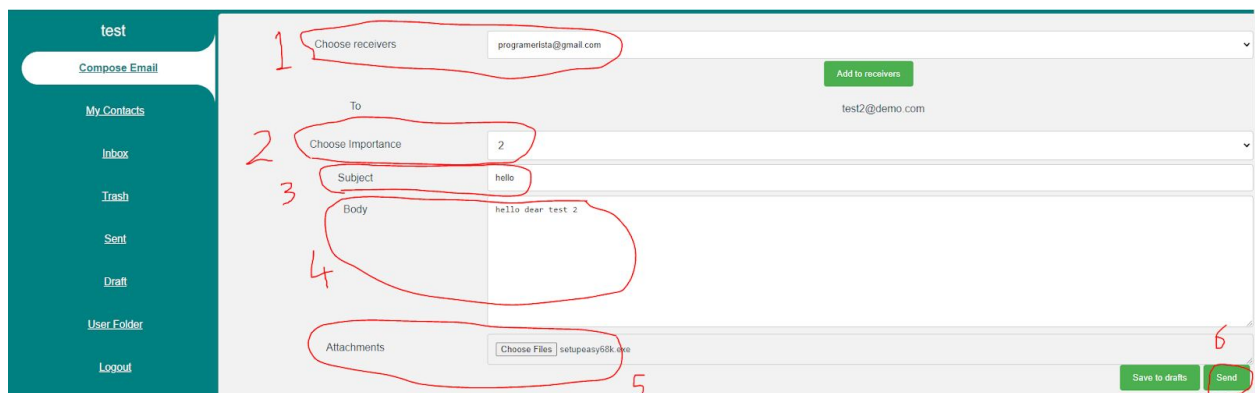
Register

Passwords do not match

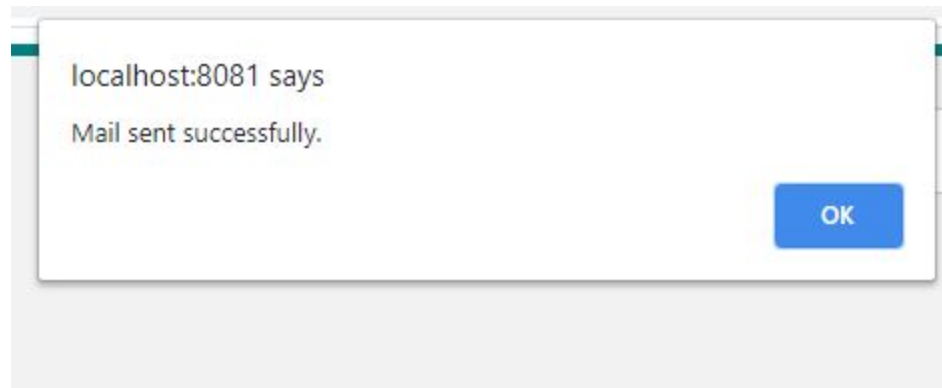
- When sign in:



- Compose new mail:



1. Choose 1 or more receivers from the users in the server.
2. Choose the importance of mail.
3. Enter the subject.
4. Write the mail .
5. Select attachments (if there exist).
6. Click send. (or save to draft if you will edit it later)



Then you can find the mail in the sent folder...

test

Compose Email

My Contacts

Inbox

Trash

Sent

Draft

User Folder

Logout

Previous PageCopyDeleteNext Page

Enter filter value...

Filter by subject

Filter

Sort by subject

Ascending

Sort

Enter search value...

Search in subjects

Search

Home

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|---------|---------------|----------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994030808 | hello | test@demo.com | ["tes12@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |

- **Mails manipulating:**

Each page contains from one to 5 mails only

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)
[Draft](#)
[User Folder](#)
[Logout](#)

Previous Page
Copy
Delete
Next Page

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|--------------------------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994030808 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| <input type="checkbox"/> 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| <input type="checkbox"/> 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |
| <input type="checkbox"/> 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| <input type="checkbox"/> 1608994325754 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:05.000+00:00 | 1 |

Next page:

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)
[Draft](#)
[User Folder](#)
[Logout](#)

Previous Page
Copy
Delete
Next Page

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|----------------------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |
| <input type="checkbox"/> 1608994354489 | find me | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:34.000+00:00 | 1 |
| <input type="checkbox"/> 1608994368000 | zender | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:48.000+00:00 | 1 |

1- **Sorting**: You can sort by every mail attribute (date, sender, receivers number, subject, importance, body size,etc) and you can sort ascending and descending.

For example i will sort by subject descendingly (from Z to A):

Enter filter value.. Filter by subject Filter Sort by subject Descending Sort

Enter search value.. Search in subjects Home

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|----------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994030808 | hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| <input type="checkbox"/> 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| <input type="checkbox"/> 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |
| <input type="checkbox"/> 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| <input type="checkbox"/> 1608994325754 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:05.000+00:00 | 1 |

1. Select sort type.
2. Select arrangement.
3. Click sort.
4. Then.. **BOOM !**

test Previous Page Copy Delete Next Page

Compose Email My Contacts Inbox Trash Sent Draft User Folder Logout

Enter filter value.. Filter by subject Filter Sort by subject Descending Sort

Enter search value.. Search in subjects Search Home

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|----------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994368000 | zender | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:48.000+00:00 | 1 |
| <input type="checkbox"/> 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |
| <input type="checkbox"/> 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| <input type="checkbox"/> 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| <input type="checkbox"/> 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |

2- **Searching:** You can search through (sende, receivers, subject, mail body) or a whole search that searches in all the mentioned attributes together.

For example i will search for **t** in whole search:

| ID | Subject | Sender | Receivers | Date | Importance |
|---------------|--------------------------------|---------------|---------------------------------------------------------------|-------------------------------|------------|
| 1608994368000 | zender | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:48.000+00:00 | 1 |
| 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |
| 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |

1. Enter searching value.
2. Select search type.
3. Click search.
4. Then.. **BOOM !**

| ID | Subject | Sender | Receivers | Date | Importance |
|---------------|--------------------------------|---------------|---------------------------------------------------------------|-------------------------------|------------|
| 160899430808 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |
| 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |

3- Filtering: You can filter through (sender or subject)

For example i will filter for **hello** in whole subject:

The screenshot shows the email filtering interface. A red circle labeled '1' highlights the input field containing 'hello'. A red circle labeled '2' highlights the dropdown menu with 'Filter by subject' selected. A red circle labeled '3' highlights the 'Filter' button. Below the input field, there is a 'Whole search' dropdown and 'Search' and 'Home' buttons. The table below shows the filtered results.

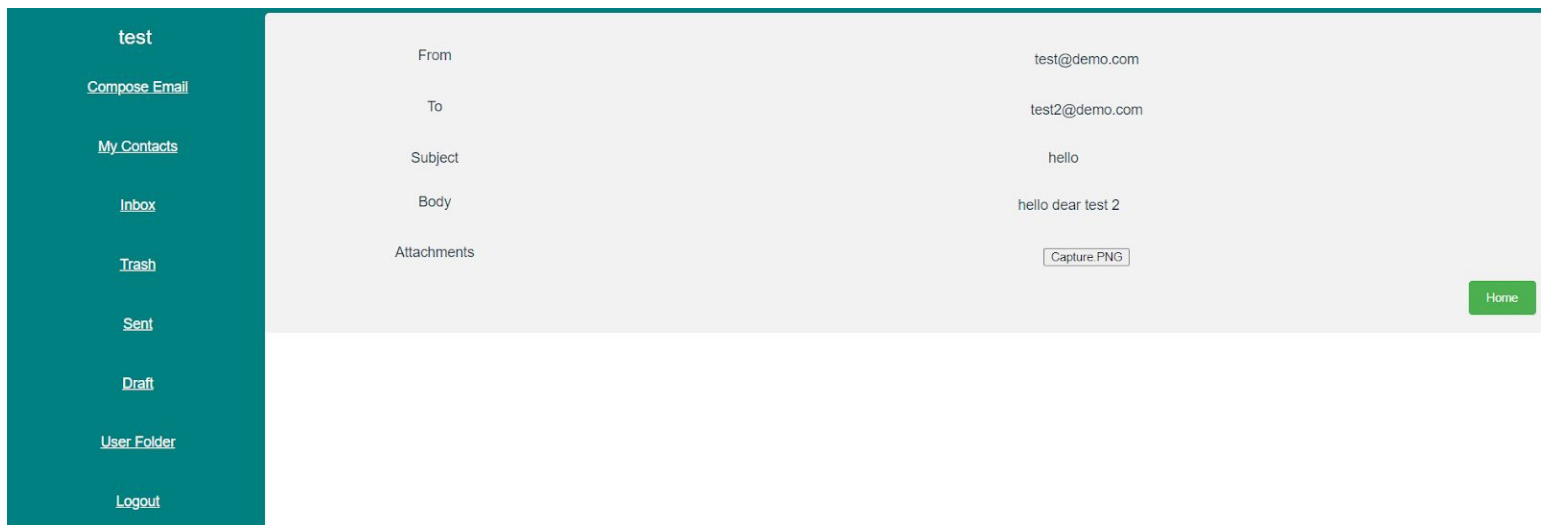
| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|--------------------------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994030808 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| <input type="checkbox"/> 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| <input type="checkbox"/> 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |
| <input type="checkbox"/> 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| <input type="checkbox"/> 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |

1. Enter filter value.
2. Select filter type.
3. Click filter.
4. Then .. **BOOM !**

The screenshot shows the email interface. On the left is a sidebar with links: test, Compose Email, My Contacts, Inbox, Trash, Sent, Draft, User Folder, and Logout. The main area shows the filtering interface with the same controls as the previous screenshot. The table below shows the filtered results.

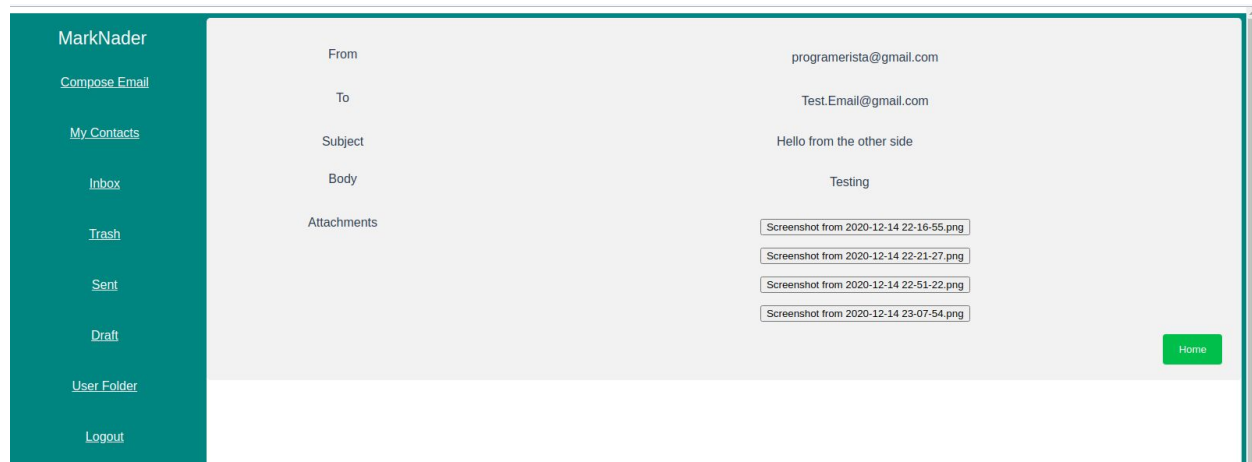
| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|-----------------------|---------------|----------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994030808 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |

- View mail:

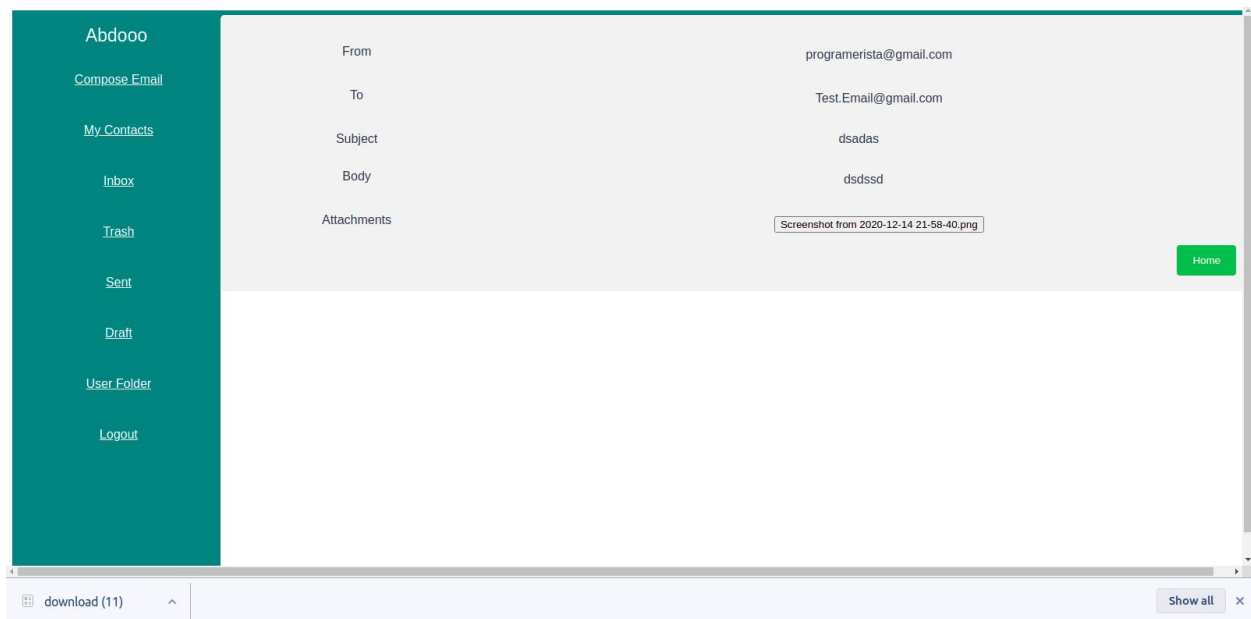


- Accessing attachments:

Viewing an email with attachments:

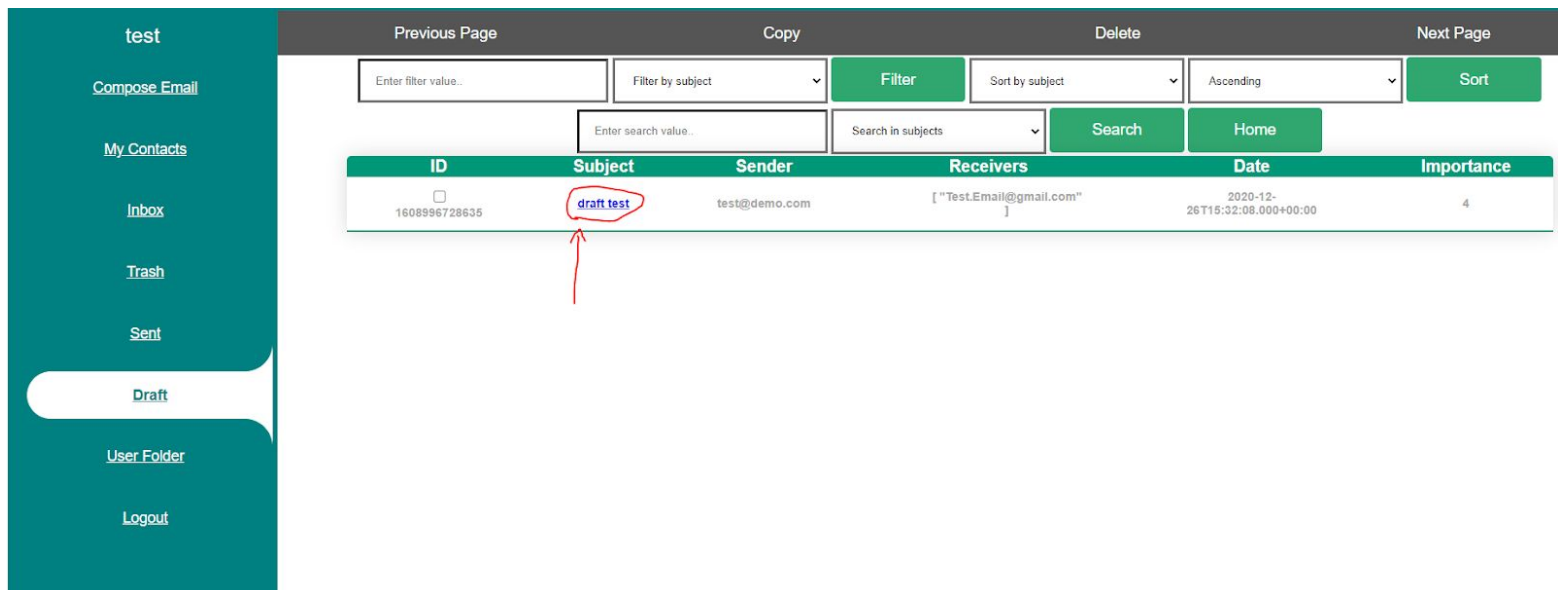


When you click in an attachment ==> **The attachment file is downloaded**



- **Edit mail from draft:**

Click on the mail from draft



Then it open in the compose window

test

Compose Email

My Contacts

Inbox

Trash

Sent

Draft

User Folder

Logout

Choose receivers

programerista@gmail.com

Add to receivers

To

Choose Importance

1

Subject

draft test

Body

x

Attachments

Choose Files No file chosen

Save to drafts Send

- Delete and restore mails from trash:

To delete mails

test

Compose Email

My Contacts

Inbox

Trash

Sent

Draft

User Folder

Logout

Previous Page

Copy

Delete

Next Page

Enter filter value..

Filter by subject

Filter

Sort by subject

Ascending

Sort

Enter search value..

Search in subjects

Search

Home

| ID | Subject | Sender | Receivers | Date | Importance |
|---------------|----------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| 1608994030898 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| 1608994269968 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| 1608994296898 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |
| 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |

1. Select a mail or more.
2. Click delete.
3. Then .. **BOOM !**

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)
[Draft](#)
[User Folder](#)
[Logout](#)

Previous Page
Copy
Delete
Next Page

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| <input type="checkbox"/> 1608994344293 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |
| <input type="checkbox"/> 1608994354489 | find me | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:34.000+00:00 | 1 |
| <input type="checkbox"/> 1608994368000 | zender | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:48.000+00:00 | 1 |

To restore mails from trash

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)
[Draft](#)
[User Folder](#)
[Logout](#)

Previous Page
Restore
Delete
Next Page

| ID | Subject | Sender | Receivers | Date | Importance |
|---------------------------------------------------|----------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994325754 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:05.000+00:00 | 1 |
| <input checked="" type="checkbox"/> 1608994330808 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| <input checked="" type="checkbox"/> 1608994349966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| <input type="checkbox"/> 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |

1. Select mails from trash.
2. Click restore.
3. Then .. **BOOM !**

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)

Previous Page Restore Delete Next Page

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|----------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994325754 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:05.000+00:00 | 1 |
| <input type="checkbox"/> 1608994296858 | important mail | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:36.000+00:00 | 1 |

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)
[Draft](#)
[User Folder](#)

Previous Page Copy Delete Next Page

| ID | Subject | Sender | Receivers | Date | Importance |
|----------------------------------------|------------|---------------|-----------------------------------------------------------------|-------------------------------|------------|
| <input type="checkbox"/> 1608994354489 | find me | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:34.000+00:00 | 1 |
| <input type="checkbox"/> 1608994030808 | hello | test@demo.com | ["test2@demo.com"] | 2020-12-26T14:47:10.000+00:00 | 2 |
| <input type="checkbox"/> 1608994269966 | new mail | test@demo.com | ["programerista@gmail.com"] | 2020-12-26T14:51:09.000+00:00 | 1 |
| <input type="checkbox"/> 1608994312374 | test | test@demo.com | ["programerista@gmail.com", "ali@demo.com"] | 2020-12-26T14:51:52.000+00:00 | 1 |
| <input type="checkbox"/> 1608994344283 | user hello | test@demo.com | ["programerista@gmail.com", "ali@demo.com", "test2@demo.com"] | 2020-12-26T14:52:24.000+00:00 | 1 |

- **Contacts and manipulating contacts:**

To view contact click my contact tab

test
[Compose Email](#)
[My Contacts](#)
[Inbox](#)
[Trash](#)
[Sent](#)

Previous Page Add Contact Delete Next Page

| ID | Name | Emails |
|--------------------------|------------|---------------------------------------------------------|
| <input type="checkbox"/> | abdo | ["abdo@demo.com"] |
| <input type="checkbox"/> | contest | ["con@test.com", "con2@test.com", "contest@demo.com"] |
| <input type="checkbox"/> | mark nader | ["Mark.N.Gendy@gmail.com"] |
| <input type="checkbox"/> | Mina Henen | ["mina@demo.com"] |

To add contact click add new contact then enter contacts data

| Previous Page | Add Contact | Delete | Next Page |
|--------------------------|-------------|---------------------------------------------------------|-------------|
| Ascending | Sort | Enter search value.. | Search Home |
| ID | Name | Emails | |
| <input type="checkbox"/> | abdo | ["abdo@demo.com"] | |
| <input type="checkbox"/> | contest | ["con@test.com", "con2@test.com", "contest@demo.com"] | |
| <input type="checkbox"/> | mark nader | ["Mark.N.Gendy@gmail.com"] | |
| <input type="checkbox"/> | Mina Henen | ["mina@demo.com"] | |

| |
|---------------|
| test |
| Compose Email |
| My_Contacts |
| Inbox |
| Trash |
| Sent |

1. Enter contact name. (test2)
2. Enter contact e-mail. (test2@demo.com)
3. If there are more than 1 click add mail then enter the next one. (add another email)
4. Click add contact. (Add To My Contact)

1. Enter contact name.
2. Enter contact e-mail.
3. If there are more than 1 click add mail then enter the next one.
4. Click add contact.

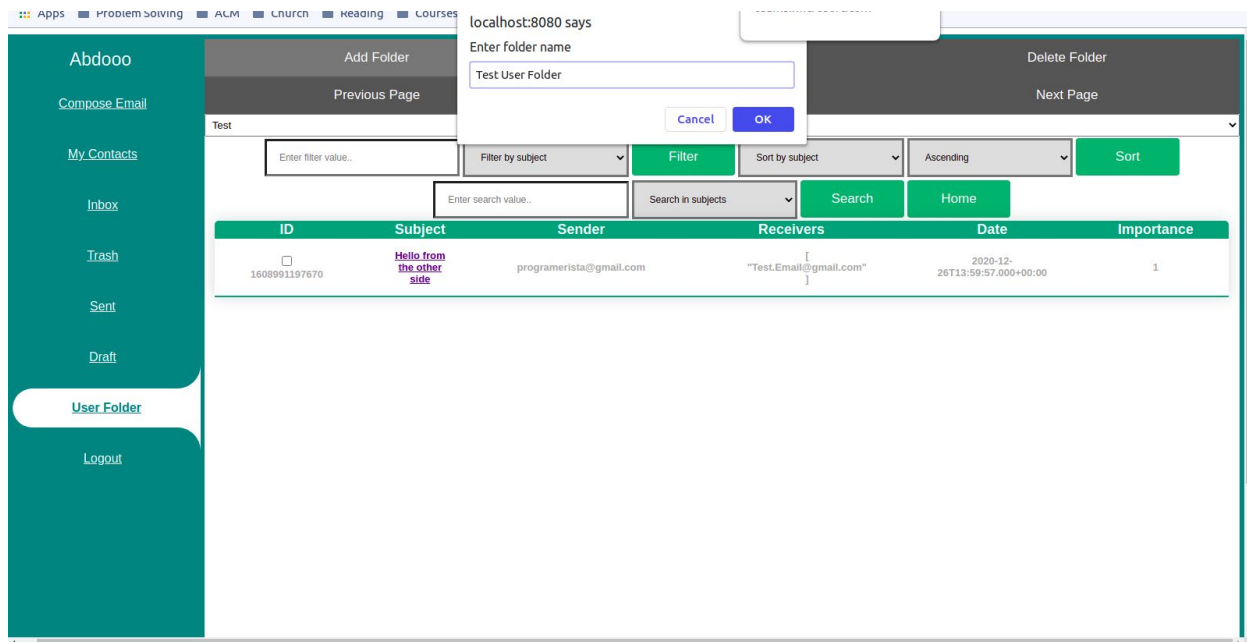
You can also sort contact according to contact names ascending from A-Z or descending Z-A and search through contact names as mentioned in sorting and searching in mails manipulation previously.

And you can delete contact as mentioned in deleting emails but without restore.

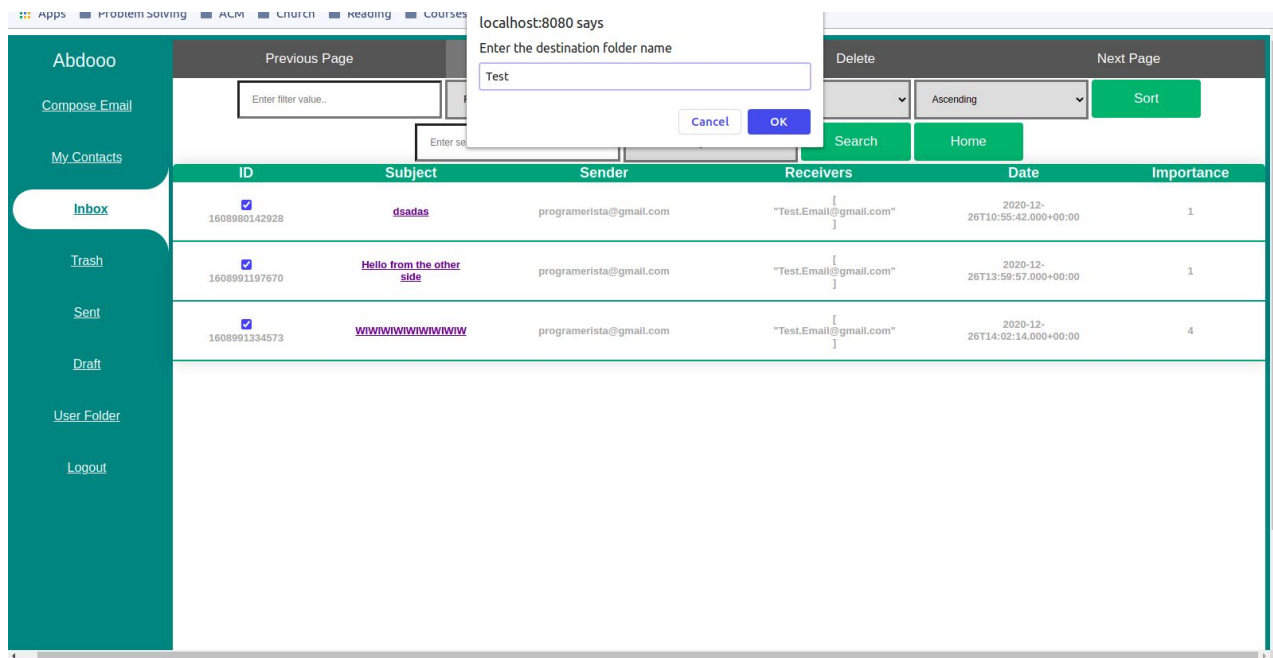
To edit contact click on it then enter new data.

• User Folders Manipulating:

First you can add a user folder and enter the name of folder



Then you can select and copy emails from the essential folders (Inbox, Sent, Draft) to the folder you created



Finally you will get the copied emails in your folder

The screenshot shows the Abdooo email interface. On the left is a teal sidebar with navigation links: Compose Email, My_Contacts, Inbox, Trash, Sent, Draft, User Folder (highlighted), and Logout. The main area has a dark header with buttons: Add Folder, Previous Page, Move, Rename Folder, Delete, and Next Page. Below the header is a search bar with a dropdown menu showing 'Test'. There are filters for 'Filter by subject' and 'Sort by subject', and buttons for 'Filter', 'Search', and 'Home'. A table displays a list of emails with columns: ID, Subject, Sender, Receivers, Date, and Importance.

| ID | Subject | Sender | Receivers | Date | Importance |
|---------------|-------------------------------------------|-------------------------|------------------------|-------------------------------|------------|
| 1608991334573 | wwwwwwwwwwww | programerista@gmail.com | "Test.Email@gmail.com" | 2020-12-26T14:02:14.000+00:00 | 4 |
| 1608980142928 | dsadas | programerista@gmail.com | "Test.Email@gmail.com" | 2020-12-26T10:55:42.000+00:00 | 1 |
| 1608991197670 | Hello from the other side | programerista@gmail.com | "Test.Email@gmail.com" | 2020-12-26T13:59:57.000+00:00 | 1 |

You can delete the user folder or rename it

The screenshot shows the Abdooo email interface with the 'Rename Folder' dialog box open. The dialog has a text input field labeled 'Enter folder new name' with the value 'test'. There are 'Cancel' and 'OK' buttons. The background interface is dimmed, showing the same sidebar and header as the previous screenshot.