

# Personal Organizer and Management Website - LifeSync

## Team Members

Mahima Borah (mahimab)  
Sambhav Jain (sambhavj)  
Zhichen Li (zl3)

## Product Backlog:

1. GUI of pages
  - a. Design (Unified CSS/Bootstrap)
  - b. Implementation of homepage
2. OAuth (Google)
  - a. Backend implementation
3. To-Do Lists
  - a. Create & Delete (Layout)
  - b. Sharing
4. Note making
  - a. Implement User designed categories
  - b. Create & Delete
  - c. Sharing
5. Calendar Integration (Google Calendar API)
  - a. Retrieve data from GC and display them (Day, Week, Month)
  - b. (option) Adding & removing
6. Expense Tracking
  - a. Implement predefined categories of expense
  - b. Visualization
  - c. Payment Notification/Reminder
7. Deploy to AWS
8. Polishment & Add-Ons

## Product Backlog (with description):

### 1. User Interface (UI):

UI/UX of pages: Designing the graphical representation of each page.

Description: This refers to the overall layout and elements that will be visually presented on each webpage.

Design (Unified CSS/Bootstrap):

Description: Ensure the design across all pages is consistent, clean and easy to use using Bootstrap.

Implementation of homepage:

Description: Building out the main landing page where users are first introduced to the application.

## 2. User Authentication:

### OAuth (Google):

Description: Implement Google's OAuth system to allow users to sign in using their Google accounts.

### Backend implementation:

Description: Set up and ensure the backend systems (like databases and servers) are working as expected.

## 3. To-Do Lists:

### Create & Delete (Layout):

Description: Allow users to create new to-do lists and delete existing ones.

### Sharing:

Description: Enable users to share their to-do lists with others.

## 4. Note making:

### Notes:

Description: Let users add notes or memos to their tasks or lists.  
Implement User designed categories.

### Create & Delete:

Description: Users can create and delete custom categories for organizing their tasks.

### Sharing:

Description: Users can share their custom categories with others.

## 5. Calendar and Appointments:

### Calendar Integration (Google Calendar API):

Retrieve data from Google Calendar and display them (Day, Week, Month):

Description: Sync with Google Calendar to fetch and display events on daily, weekly, and monthly views.

### Summarized View:

Description: We can show the user the total number of meetings he has in that particular day, week or month.

### Adding & removing (optional) :

Description: Optionally, let users add or remove events directly from the application which will also update in their Google Calendar.

## 6. Expense Management:

### Expense Tracking:

Implement predefined categories of expense:

Description: Users can track their expenses under predefined categories like 'Food', 'Travel', etc.

Visualization:

Description: Provide graphical visualizations (like pie charts or bar graphs) of the user's expenses.

Payment Notification/Reminder:

Description: Notify users of upcoming payments or due dates.

7. Deployment & Refinements:

Deploy to AWS:

Description: Make the website live by deploying it on the AWS cloud platform.

Polishment & Add-Ons:

Description: Further refine and polish the website, potentially adding more features as they are developed and Improvement in design.

## Sprint Plan

**Sprint #1 Backlog:**

**Product Owner:** Zhichen Li (zl3)

Task	Owner	Status
Wireframes	Sambhav	Done
Homepage HTML	Sambhav	Done
OAuth backend setup	Mahima	Done
OAuth integrate with homepage	Mahima	Done
To-Do List - Create and Delete	Zhichen	Done
To-Do List - Sharing	Zhichen	Done

## Data Models:

### (1) Feature: to-do list

```
class ToDoList(models.Model):
    users = models.ManyToManyField(User)
    items = models.ManyToManyField(ToDoItem)
    name = models.CharField(max_length=255, default='My To-Do List')
    date = models.DateTimeField(default=timezone.now)
    description = models.CharField(max_length=255, default='')
    key = models.BinaryField() # This is for sharing
```

```
class ToDoItem(models.Model):
    users = models.ManyToManyField(User)
    deadline = models.DateTimeField()
    item = models.CharField(max_length=255)
    status = models.BooleanField(default=False)
```

### (2) Feature: Note making

```
class NoteCategory(models.Model):
    users = models.ManyToManyField(User)
    notes = models.ManyToManyField(NoteItem)
    category_name = models.CharField(max_length=255, default='My Notes')
    created_time = models.DateTimeField()
    description = models.CharField(default='')
```

```
class NoteItem(models.Model):
    users = models.ManyToManyField(User)
    created_time = models.DateTimeField()
    latest_update_time = models.DateTimeField()
    title = models.CharField(max_length=255)
    content = models.CharField()
```

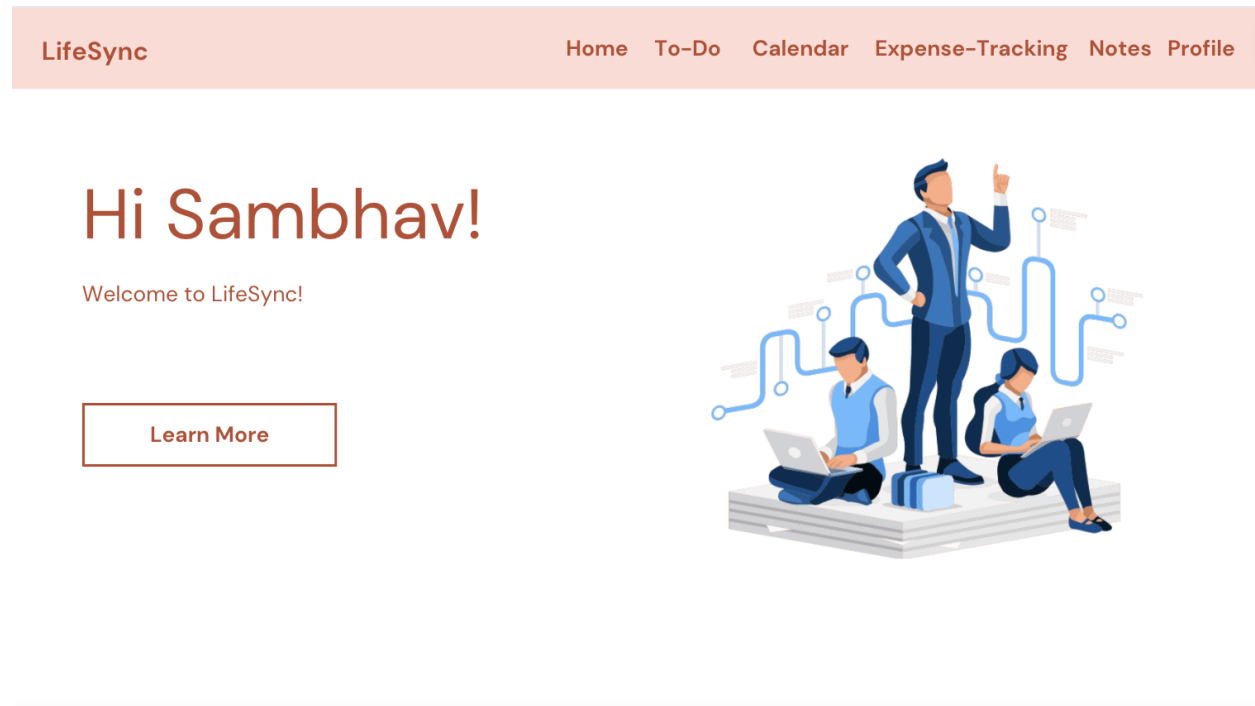
### (3) Feature: Expense Tracking

```
class Category(models.Model):
    name = models.CharField(max_length=255, unique=True)

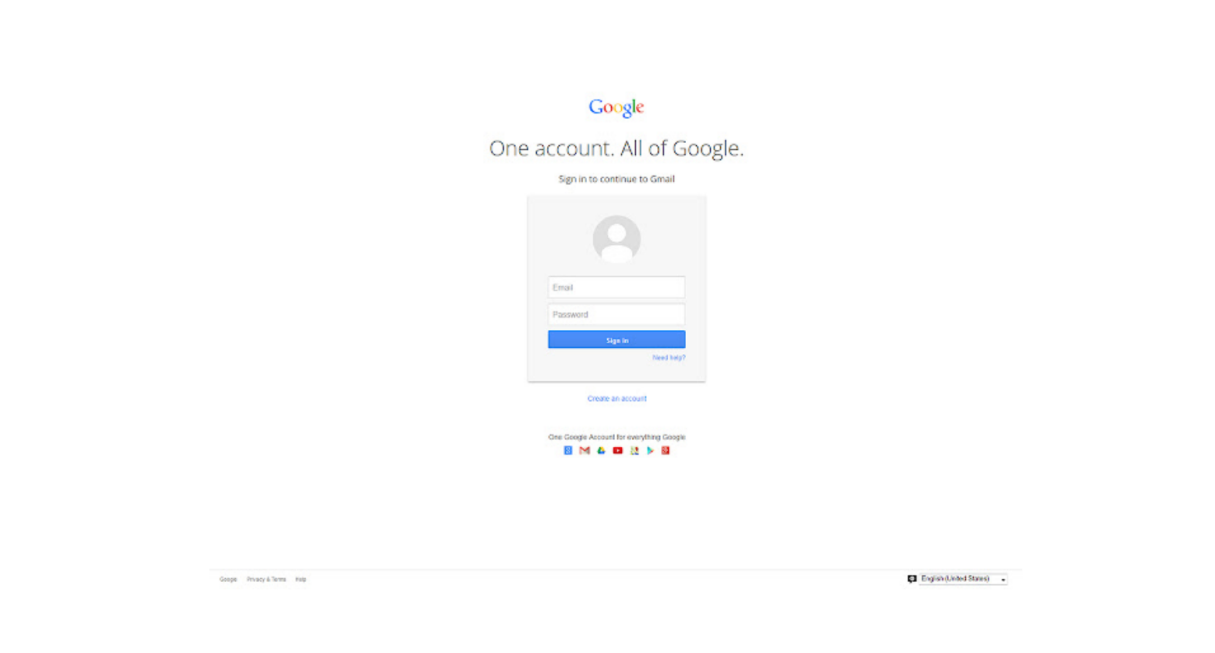
class ExpenseItem(models.Model):
    users = models.OneToOneField(User)
    created_time = models.DateTimeField()
    expense_title = models.CharField(max_length=255)
    amount = models.CharField(max_length=255)
    description = models.CharField(null=True, blank=True)
    category = models.OneToOneField(Category)
```

## Wireframes/Mockups:

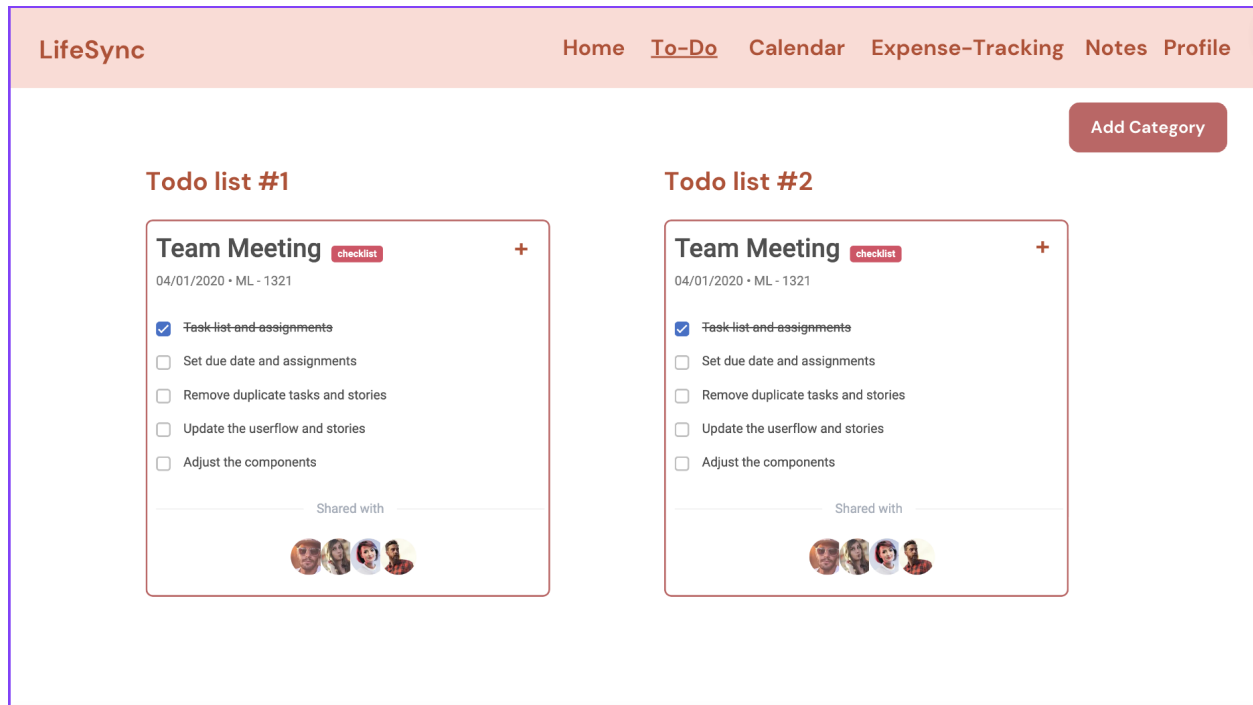
(1) Homepage: This is the landing page of our website lifesync



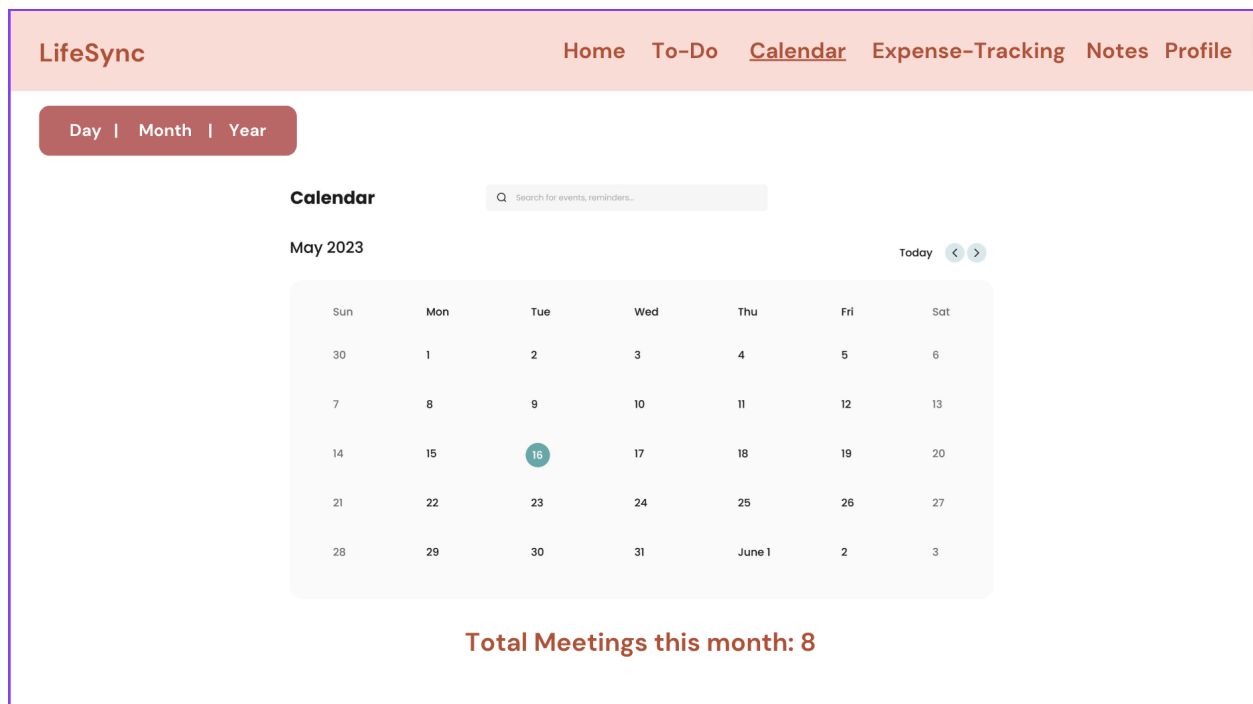
(2) Login Page: This is the google oAuth screen



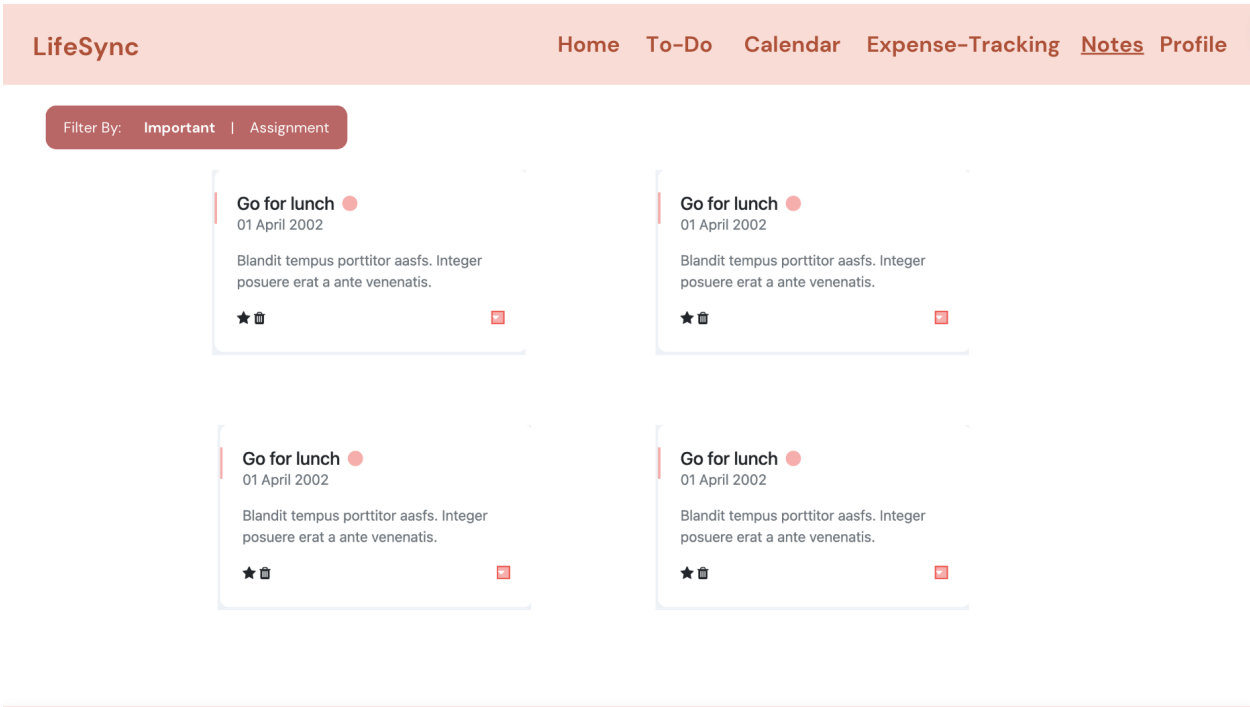
(3) To-do list: This page has the contents for our to-do list feature



(4) Calendar: This page displays the users google calendar



(5) Note Making: This page has contents for our note making feature



(6) Expense Tracking: This page

