



# Plan.ly

CS 673  
Team 1 - UNO  
Iteration 3



## Team Overview

- Team **UNO**
- Working to build Plan.ly, an event planning tool

Name	Role
George Wright	Team Leader
Vibhu Bhatia	Backup Team Leader
Aysha Zenab Kenza	Security Leader
Karen Sommer	Design/Implementation Leader
Zach Schandorf-Lartey	Configuration Leader
Chris Kulig	Requirements Leader
Matt Dowding	Quality Assurance Leader



# Content

1. Project management
2. Requirement analysis
3. Design
4. Implementation
5. Testing
6. Security
7. Deployment
8. Demo

# Project management

---



# Management Strategy

George Wright  
Team Leader

- Management style
- Meetings
- Training/pair programming
- Use of PivotalTracker for management



# Management Style

- *[Attempted to]* adhere to a **agile scrum** methodology with a working product to show at the end of each iteration
- Given team schedules:
  - No daily stand-up
  - Combined sprint planning and retrospective
- Supplement meetings with technical workshops and pair programming

**Iteration 0**

*Technical workshops about  
Django/React*

**Iteration 3**

*Pair programming (2-3  
people) for specific stories*



# Meetings

- Hosted 25 group meetings
- 30 pages of meeting minutes
- Typically Saturdays or Sundays at 2 PM
- Numerous smaller breakout groups for pair programming or troubleshooting
- Added 1-2 meetings during weeks before iteration presentations

*Sprint (iteration)  
planning/retrospective*

*Technical workshop,  
testing, or debugging*

*Non-technical (documents,  
presentation, etc.)*



# Use of PivotalTracker in Meetings

- Mainly during meetings immediately leading up to or following presentations
- Use Pivotal to drive group discussion of:
  - Bug review
  - Metric update
  - Story cleanup
  - Work assignment



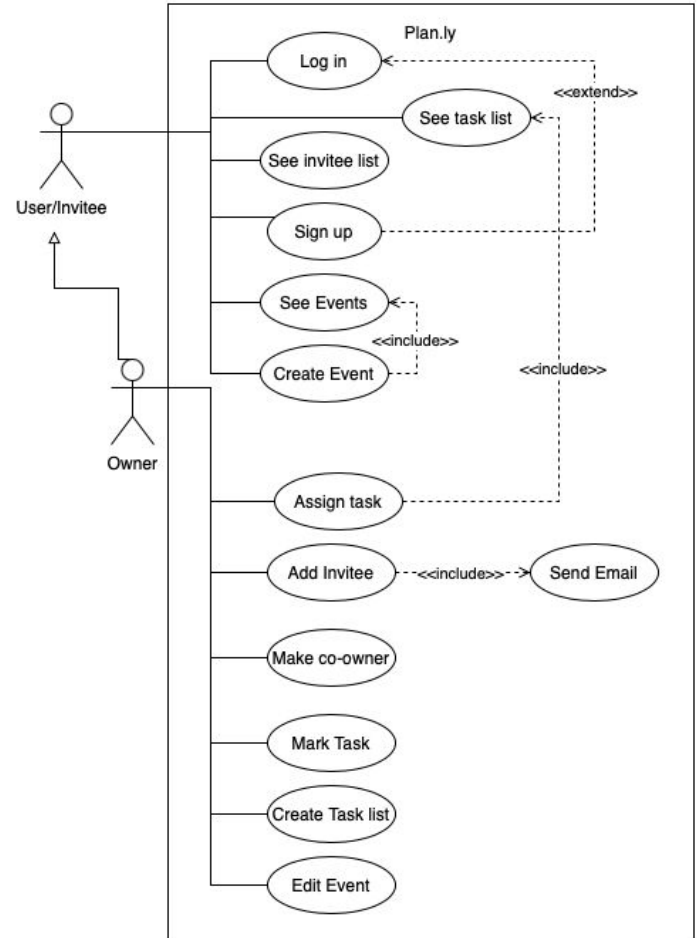


# Requirement analysis

---

# Plan.ly - Essential Stories

- Sign Up
- Log In
- Create a Event
- View Feed of Events
- “Owner” inherits from “User/Invitee”
- Final product use case diagram design



# Use Case Description



<b>Name:</b>	Add an Event	<b>Initiating actor:</b>	User/Invitee
<b>Entry condition:</b>	→ The user is logged into Planly → The user is on the dashboard page	<b>Exit condition:</b>	→ The user becomes an Owner of the created event. → The user receives display information about the error.
<b>The flow of events: &lt;Basic Flow&gt;</b>			
<b>User/Invitee Steps</b>		<b>Planly steps</b>	
1. Clicks on Create event  3. Input information about the event		2. Displays the pop up for input information 4. Validates inserted information 5. Create a new event and save it into the database 6. Displays the dashboard page with the new event	
<b>The flow of events: &lt;Alternative Flow&gt;</b>			
<b>User/Invitee Steps</b>		<b>Planly steps</b>	
1. Input information about the event [Invalid information]		[Invalid information] Displays the failure.	

## Pivotal: Starting Out

- Set detailed guide on functional expectations through user stories using Given, When, Then to describe what a completed story looks like.

A screenshot of a Pivotal user story card. The card has a title 'Log in' in a search bar at the top. Below the title, there are icons for ID, a link, a clock, a trash can, and a 'Collapse' button. The card is divided into several sections: 'STATE' with a status of 'Accepted on 3 Apr 2021', 'REVIEWS' with a '+ add review' link, 'STORY TYPE' set to 'Feature', 'POINTS' set to '2 Points', 'REQUESTER' set to 'Zach Schandorf-Lar...', 'OWNERS' with 'ZS' and 'WR' avatars and a '+', and 'FOLLOW THIS STORY' with '(3 followers)' and a checkmark. At the bottom of this section, it says 'Updated: 3 Apr 2021, 02:25pm'. Below this is a 'BLOCKERS' section with a '+ Add blocker or impediment' button. The bottom section is 'DESCRIPTION', which contains the text: 'As a web site/user I want to be able to log into my account to securely gain access to my resources. -2H'.

A screenshot of the Pivotal activity feed. At the top, it says 'ACTIVITY' and 'Sort by Oldest to newest'. There are two activity items. The first item is from user '@wrightgd' (WR) and says: 'Given I am a website user, when I visit the Plan.ly homepage, then I see a login form I can use to enter my valid email address and password.' Below this text are links for 'Reply', 'React', and 'Copy link', followed by the timestamp 'Mar 7, 1:11 pm' and the word 'edited'. The second item is also from user '@wrightgd' (WR) and says: 'Given I am a website user, when I attempt to login to Plan.ly with a set of credentials known to be valid, then I am redirected to my user dashboard/feed page as a logged-in user.' Below this text are links for 'Reply', 'React', and 'Copy link', followed by the timestamp 'Mar 7, 1:10 pm'.



## Pivotal: Starting Out

- Developer defined tasks are crucial to estimating development time and the pace of

### TASKS (7/7)

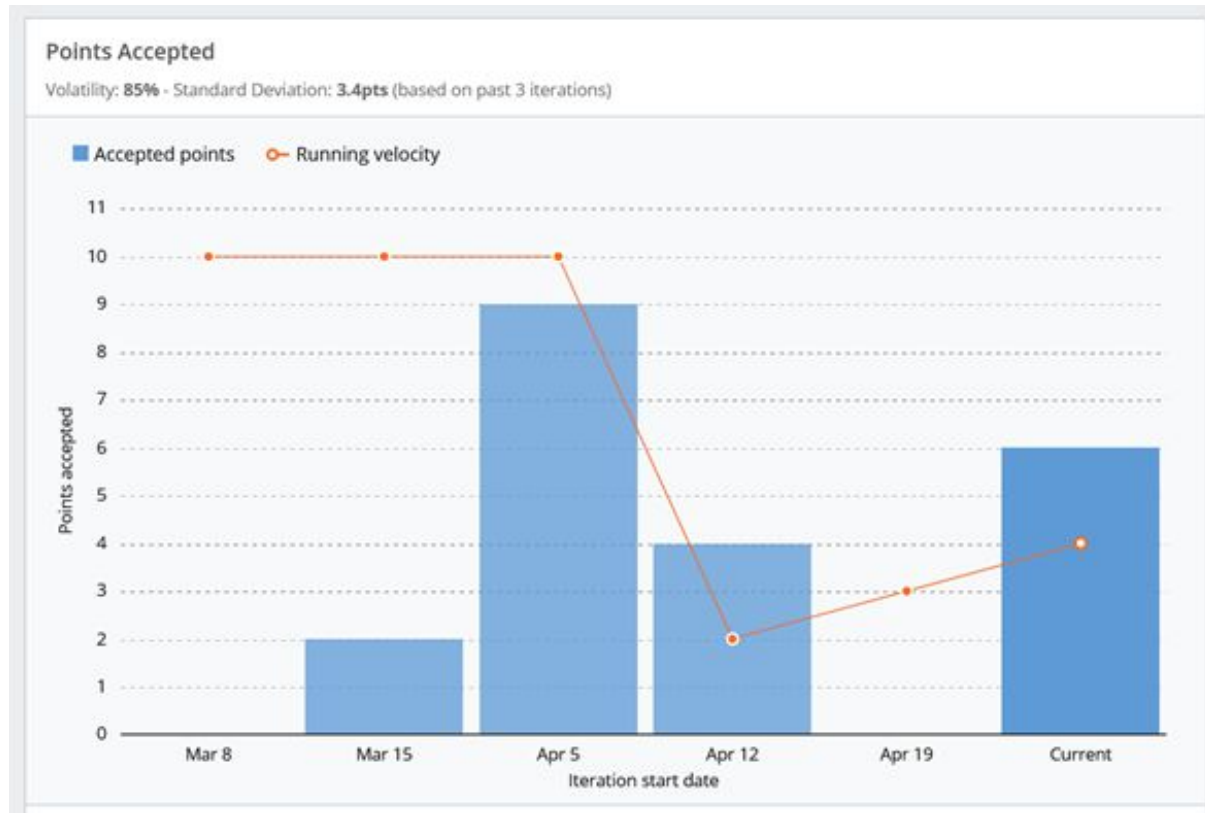
- ☒ Create authentication flow - 2h
- ☒ Create a login/signup component - .5h
- ☒ create the logic to check if the user logging in already has an account in the django database - 1h
- ☒ write user tests - 1h
- ☒ test the log in flow -1h
- ☒ if the user does not exist, create logic to create user account in mongo database with the provided name/email/password -3h
- ☒ if the user does exist, create logic to send them to the dashboard component - 1h

## Pivotal: Making Progress

- We used Pivotal to
  - Guide development
  - Adjust our direction if necessary
  - Log bugs

★ =	Sign up (ZS, WR) essential, user authentication/registration /profile	<input type="checkbox"/>
★ —	View my current private events (WR) basic event management, essential	<input type="checkbox"/>
★ ≡	Create a private event (WR) basic event management, essential	<input type="checkbox"/>
🚫	When app is left unattended until session expiration, user is not gracefully redirected to a login screen (ZS)	<input type="checkbox"/>
🚫	Cannot query by ID on event model (WR)	<input type="checkbox"/>

# Results



# Results

Accepted Story Type Composition







## Results

Config is Frontloaded

## View feed of events

TASKS (6/6)



Create user class - .5h



Create the user in the database - .5h



Create the event class - .5h



Create an event table in the database - .5h



Create the React UI that will display the users event information - 1.5h



Unit test - 1.5h

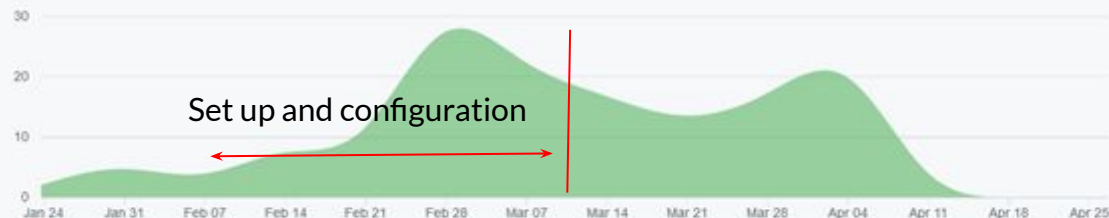
# Results

## Config is Frontloaded

Jan 24, 2021 – Apr 27, 2021

Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts



Accepted Story Type Composition

Features Bugs Chores



# Design

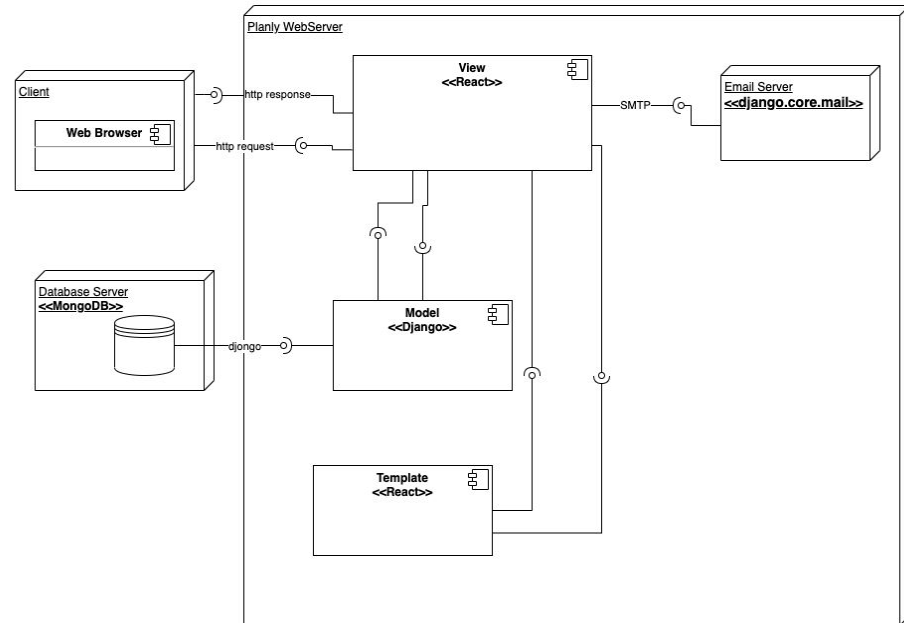
---

# Software Architecture

## -Component diagram

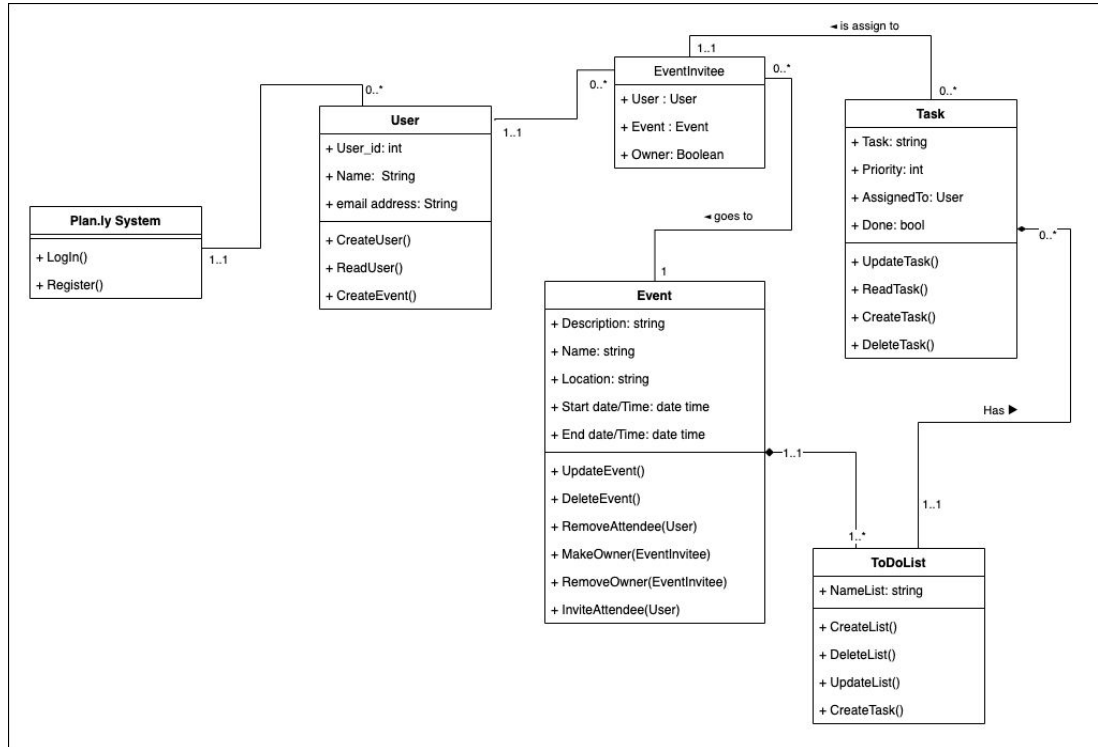
### Architectural Pattern

- Django architecture
- MTV pattern
  - Modified MVC.



# Software Architecture

## -Class diagram



## Classes

1. Plan.Iy System
2. User
3. Event Invitee
4. Event
5. ToDoList
6. Task

## Classes and Methods

# Database

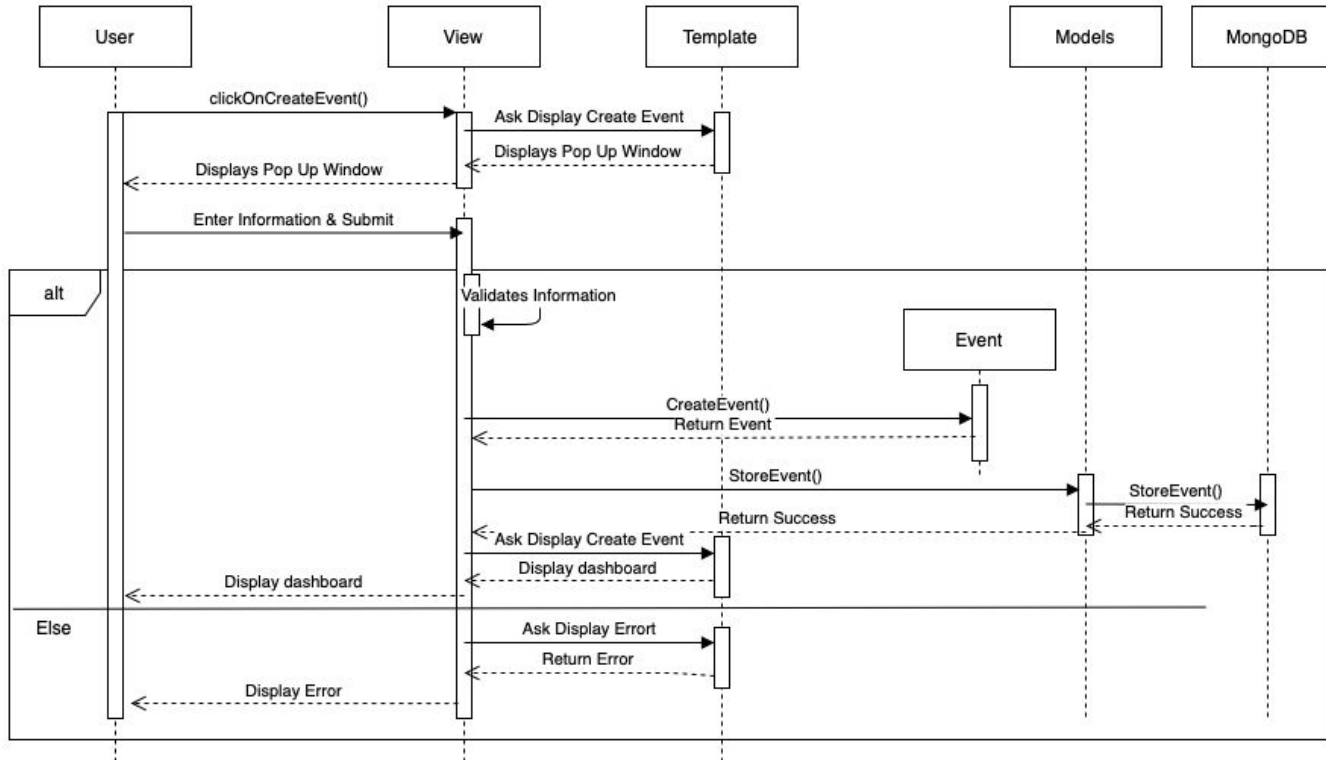
- Plan.ly uses NoSQL Database (**mongoDB**)
- Django Provides a mongoDB interface, **Djongo** which allows easy integration

Event Class

```
{
  "_id" : <DB generated id>,
  "Name" : "My Sample Event",
  "Description" : "This is the sample description for my awesome event. Please be there. I need help...",
  "Location" : "Boston, Massachussetts",
  "Start_date" : Jan 26 2021 10:20:00 AM,
  "End_date" : Jan 30 2021 10:20:00 AM,
  "Participants" : [Event_invitee1_id, Event_invitee2_id, Event_invitee3_id, Event_invitee4_id...],
  "Owner" : Event_invitee_id,
  "Lists" : [todoList_1_id, todoList_2_id, todoList_3_id, todoList_4_id..],
}
```

# Sequence Diagram

## -Add Event



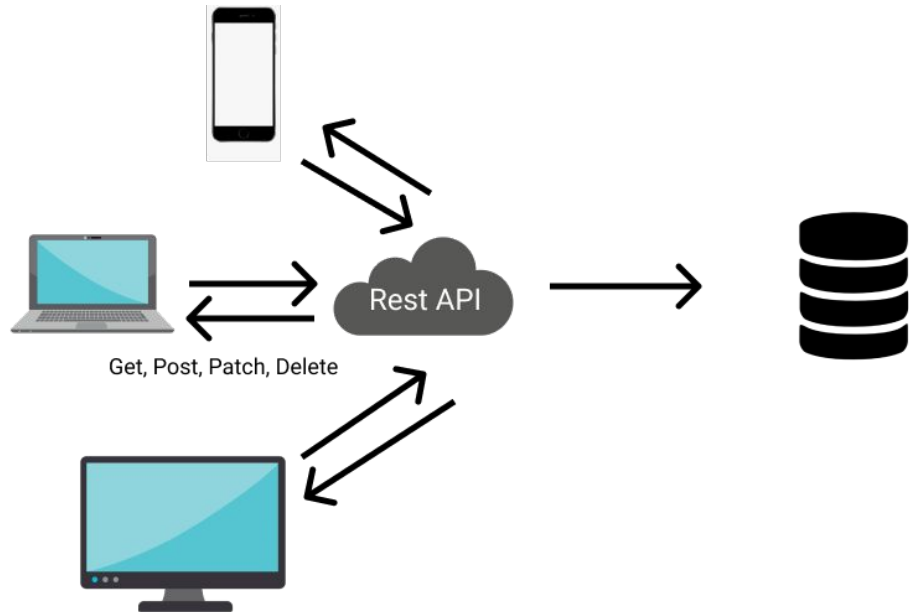
# Implementation

---



# System Architecture

- Stateless architecture
- Supported across multiple devices.
- Highly Scalable.
- Frontend-backend segregation.



# Backend : API

localhost:8000/docs#tag/api

Search...

Authentication

api

- GET api\_event\_list
- POST api\_event\_create
- GET api\_event\_read
- PUT api\_event\_update
- PATCH api\_event\_partial\_update
- DELETE api\_event\_delete
- GET api\_event\_refresh
- GET api\_eventintviee\_list
- POST api\_eventintviee\_create
- POST api\_eventintviee\_toggle
- DELETE api\_eventintviee\_uninvite
- POST api\_eventintviee\_validate
- GET api\_eventintviee\_read

api

api\_event\_list

AUTHORIZATIONS Basic

Responses

— 200

api\_event\_create

AUTHORIZATIONS Basic

Responses

```
class TaskView(viewsets.ModelViewSet):
    def list(self, request):
        # /api/tasks/ (get)
    def create(self, request):
        # /api/tasks/ (post)
    def retrieve(self, request, pk=None):
        # /api/tasks/<task_id> (get)
    @action(detail=True, methods=["get"])
    def refresh(self, request, pk=None):
        # /api/tasks/<task_id> (get)
    @action(detail=True, methods=["post"])
    def changeOwner(self, request, pk=None):
        # /api/tasks/<task_id> (post)
    def destroy(self, request, pk=None):
        # /api/tasks/<task_id> (delete)
```

# Backend : Database (NoSQL)

```
_id: ObjectId("6081e13bcbff4d74a9944e10")
name: "Make Eggs"
priority: 1
└ invitee_assigned: Object
  _id: ObjectId("6081e0eccbff4d74a9944e0e")
  user_id: 36
  isOwner: true
  event_id: 139
completed: false
list_id: "6081e12ecbff4d74a9944e0f"
```

## Some Features of Database:

- Added redundancy (increased in Space complexity).
- No expensive joins needed (fast response - api calls).
- Embedded documents saved as separate files for separate retrieval.
- Nesting allows for arbitrary levels of storage.

```
_id: ObjectId("60876684e02398f215ab8214")
name: "test"
description: "test"
> creator: Object
└ tasks: Array
  > 0: Object
  > 1: Object
  └ 2: Object
    _id: ObjectId("6089784ba75ea8617b9fe67a")
    name: "woohoo"
    priority: 1
  > invitee_assigned: Object
    completed: false
    list_id: "60876684e02398f215ab8214"
```

# Frontend : Components

- Html page is divided into components.
- Some components are initialized after certain conditions go through.
- Information is stored in state variable
- As Parent or child component gets updated the updated information is passed through all the components that need that information so that updated info is reflected globally.
- Dashboard has the most expensive operation.
- Pages fetch information independently (makes execution faster)

```
/**
 * Class representing the Dashboard once user logs in.
 * @extends Component
 */
export default class Dashboard extends Component {
  /**
   * Create a Dashboard
   * @param {string} name - The ???
   * @param {string} username - The name of the person logged in.
   * @param {string} items - The list of items in that person's account.
   */
  constructor(props) {
    super(props);

    this.state = {
      username: '',
      user_id: -1,
      logged_in: !!localStorage.getItem('token'),
      userEventList: [],
      eventFormComponent: null,
      calendarEvents: [],
      allTaskComponent: (
        <span>
          <Col>
            <Spinner animation="border" variant="danger" />
          </Col>
          <Col>
            Loading your tasks.
          </Col>
        </span>
      ),
      recUpComponent: (
        <span>
          <Col>
            <Spinner animation="border" variant="primary" />
          </Col>
          <Col>
```

# Frontend : Components

The screenshot displays the Plan.ly application interface. The top navigation bar is blue with the Plan.ly logo and a user profile icon. The left sidebar contains links for 'Home' and 'Events'. The main content area is titled 'Updated Event' and includes details about the event location and description. A section titled 'Who's coming?' lists invitees. Below this, there is a grid of task lists, each with a title, a description, and a list of tasks. A red box highlights the 'Edit Event Details' button in the top right corner. Another red box highlights the 'Add Invitee' button in the 'Who's coming?' section. A third red box highlights a grid of task lists, each with a title, a description, and a list of tasks. A fourth red box highlights the '+ New Task List' button in the bottom right corner.

Plan.ly

Home

Events

## Updated Event hosted by Me!

Updated Location: 2022-11-11 00:00:00 - 2022-11-22 00:00:00

Updated Description

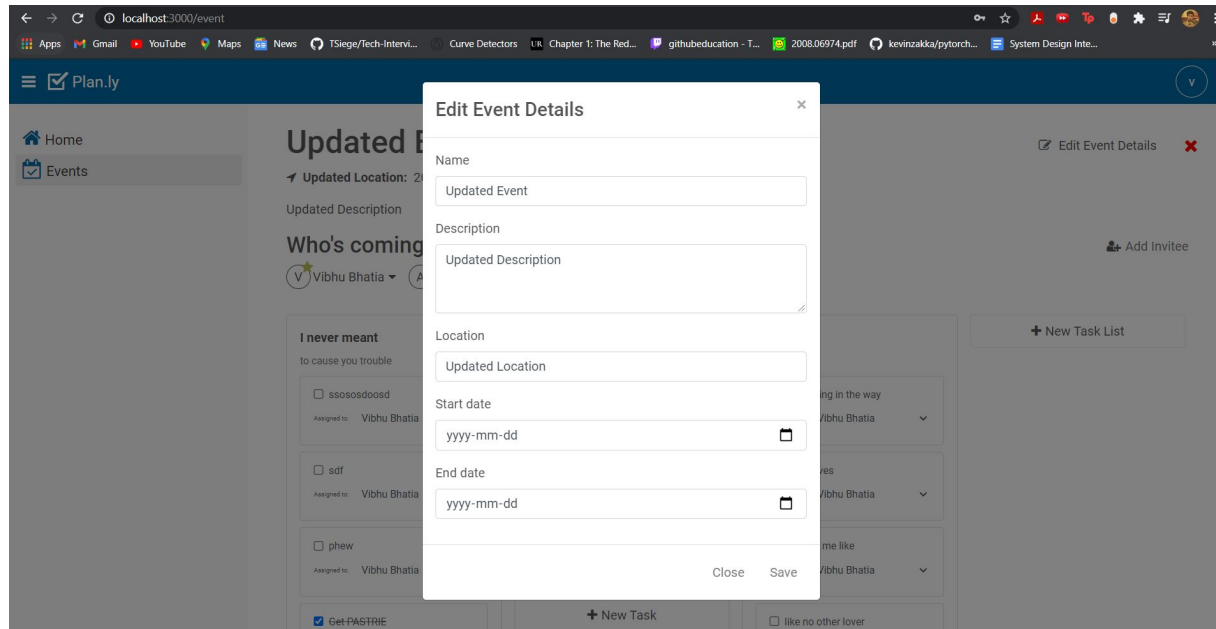
### Who's coming?

Vibhu Bhatia Aysha Zenab Kenza

+ New Task List

I never meant	test	something
to cause you trouble	test	ssddf
<input type="checkbox"/> ssososoosd Assigned to: Vibhu Bhatia	<input type="checkbox"/> hyeeee Assigned to: Vibhu Bhatia	<input type="checkbox"/> something in the way Assigned to: Vibhu Bhatia
<input type="checkbox"/> sdf Assigned to: Vibhu Bhatia	<input type="checkbox"/> ggg Assigned to: Vibhu Bhatia	<input type="checkbox"/> she moves Assigned to: Vibhu Bhatia
<input type="checkbox"/> phew Assigned to: Vibhu Bhatia	<input type="checkbox"/> woohoo Assigned to: Vibhu Bhatia	<input type="checkbox"/> attracts me like Assigned to: Vibhu Bhatia

# Frontend : Components





## Refactoring Case Study:

1. Event page displayed all the information related to an event.
2. Edit event modal could be used in similar way as the create event modal in home page.
3. As we started delegating more duties to group members, everyone needed to edit state variables and not affect or break webapp for others.
4. State should only hold the most important information, leave the rest to either rendering or external functions (render called every time state is updated).
5. Updates in child component not reflected in the parent component or sibling components.
6. Every Page now independently fetches for information.

# Testing

---



# Unit Testing: python unit test

**Unit Tests.** For unit testing, we have made use of the Django built-in unit testing mechanism in tests.py. In this file, we have checks to verify that we can create and delete model records with the expected successes and validation exceptions being returned by the back-end. Example tests include:

- Checking to make sure that the id field of the User model cannot be assigned by a developer; only the back-end
- Testing that Event Invitee records can be created and tied to an event/user using the respective ID attributes of each
- Testing behavior when task records are given extremely long names/descriptions (exceeding the validation requirements)
- Checking that no ValidationError is raised when an Event record is created with all required field filled in correctly.

```
(planly) D:\github\ms_cs\Spring-2021\software engineering\BUMETCS673S21T1>python manage.py test
System check identified no issues (0 silenced).
```

```
.....
```

```
-----
Ran 7 tests in 0.914s
```

```
OK
```



# System Testing: selenium

**System Tests.** Our system testing approach relies upon Selenium, which we have imported through its Python module. The system testing is designed specifically around the requirements we have stored in PivotalTracker: in the spreadsheets referred-to below, we specifically map each test case to the requirements it exercises. We aimed to automate each and every single one of our test cases. By doing so, we will ensure they can be quickly run, and serve as a valuable tool to quickly test changes to our app and ensure nothing has regressed. Samples of system test cases include (but are not limited to):

- Sign up for a new Plan.ly account and ensure you can login
- Attempt to login with credentials known to be invalid and ensure a warning is given
- Sign in, navigate to an Event page, and attempt to edit the event; ensure the details are updated on the event page once “Save” is clicked on the edit form

# System Testing: requirement testing coverage sheet

Pivotal Export (4-22-2021)				Test cases						
Id	Title	Description	Labels	TC-01	TC-02	TC-03	TC-04	TC-05	TC-06	TC-07
176949922	Sign up	essential, user authentication/registration/profile	As a website user I want to be able to create an account to be able to create so that I can log into the page as an authorized user and use the page functions as necessary.8.5H							
176879523	Log in	essential, user authentication/registration/profile	As a web site/user I want to be able to log into my account to securely gain access to my resources. -2H							
176879628	Create a private event	basic event management, essential	As a website member(Signed User), I want to create a private event, so that I can share details like the description, location, and starting and ending date and time, of the event with the invitee. 6.5 H							
176879631	View my current private events	basic event management, essential	As an event owner, I want to view my current private events, so that I see how are they going - 4H							
176879638	Update my private event	basic event management, desirable	As an event owner, I want to update a private event, so that I can modify the information(description, starting and ending date, and time) of the event. 7.2H							
176953068	Go To Event From Feed	desirable, user dashboard/feed	As a logged in user on the home page when there are events available in the feed I want to be able to click on an event within the feed and go to the details page to get information about it							

# System Testing: end-to-end test cases sheet

Test Case ID	Description	Type	Objective	Procedure	Expected Results	2021-04-03 15:34:47		2021-04-04 18:58:05		2021-04-05 19:42:03	
						Results	Comments	Results	Comments	Results	Comments
TC-01	Sign up with <a href="#">Plan.ly</a> for the first time, using a valid email account	Positive Functionality	Ensure that <a href="#">Plan.ly</a> allows new (unregistered) users to sign up using a valid email account	1. Open web browser and navigate to <a href="#">Plan.ly</a> homepage 2. Click "signup" to initiate registration process 3. Choose the option to register with a Google account 4. Type in the credentials of a known email account to be used for testing 5. Click submit	1. Account is successfully created and user is redirected to the dashboard/feed page. 2. User receives an automated email (at the email address associated with the email account) confirming that the registration was successful.	P	N/A	Error: Name Error	syntax error	P	N/A
TC-02	Log in to <a href="#">Plan.ly</a> with a valid email account	Positive Functionality	Ensure that <a href="#">Plan.ly</a> allows a previously registered user to sign in with their associated email account	1. Open web browser and navigate to <a href="#">Plan.ly</a> homepage 2. Click "Sign In" button to initiate logging in 3. Choose same email account the user has previously signed up with 4. Click "submit" button	1. Successful log in with correct Google account 2. Redirect to user dashboard	P	N/A	P	N/A	P	N/A
TC-03	Logout from <a href="#">Plan.ly</a>	Positive Functionality	Ensure that <a href="#">Plan.ly</a> allows users to logout after they successfully logged in	1. Open web browser and navigate to <a href="#">Plan.ly</a> homepage. 2. Login with known email account credentials that were previously used to register on Plan.ly 3. Once logged in, click logout	1. Successfully logged out of users account 2. Return to <a href="#">Plan.ly</a> homepage	P	N/A	P	N/A	P	N/A
TC-04	Logout from Plan.ly and attempt to click back button	Negative Functionality	Ensure that on successful log out a user cannot hit the back button to see restricted content	1. Open web browser and navigate to <a href="#">Plan.ly</a> homepage. 2. Login with known email account credentials that were previously used to register on Plan.ly 3. Once logged in, click logout 4. Click the "back" browser button	1. The user is not able to return to previous restricted content 2. User redirected to <a href="#">Plan.ly</a> homepage	P	N/A	P	N/A	P	N/A
TC-05	Create a private event	Positive Functionality	Ensure that a registered user on <a href="#">Plan.ly</a> can create an event	1. Open web browser and navigate to <a href="#">Plan.ly</a> homepage. 2. Login with known email account credentials that were previously used to register on Plan.ly 3. Click on create an event 4. Verify that an event has been created	1. Successfully create an event for the user	P	N/A	Error: No Such Element Exception	failure after merge with updated dev	P	N/A

# System Testing: report generation

## Unittest Results

**Start Time:** 2021-04-27 11:02:47

**Duration:** 166.60 s

**Summary:** Total: 14, Pass: 14

__main__.AutomatedTesting	Status
test_TC_01	Pass
test_TC_02	Pass
test_TC_03	Pass
test_TC_04	Pass
test_TC_05	Pass
test_TC_06	Pass
test_TC_07	Pass
test_TC_08	Pass
test_TC_09	Pass
test_TC_10	Pass
test_TC_11	Pass
test_TC_12	Pass
test_TC_13	Pass
test_TC_14	Pass

Total: 14, Pass: 14 -- Duration: 166.60 s

# System Testing: report generation - catching bugs

## Unittest Results

Start Time: 2021-04-27 10:20:15

Duration: 121.77 s

Summary: Total: 12, Pass: 7, Error: 5

__main__AutomatedTesting	Status	
test_TC_01	Pass	
test_TC_02	Pass	
test_TC_03	Pass	
test_TC_04	Pass	
test_TC_05	Pass	
test_TC_06	Pass	
test_TC_07	Pass	
test_TC_08	Error	Hide
<p>NoSuchElementException: Message: no such element: Unable to locate element: {"method":"xpath","selector":"/html[@id='events']/div/div/div/div/div/div[2]/a/div/div/h3"} (Session info: chrome=90.0.4430.93)</p> <p>Traceback (most recent call last): File "c:\Users\Matthew\Desktop\project673\main\BUMETCS673S21T1\testing\automated_testing.py", line 155, in test_TC_08 self.driver.find_element_by_xpath("//*[@id='events']/div/div/div/div/div/div[2]/a/div/div/h3).click() File "C:\Users\Matthew\Desktop\project673\main\BUMETCS673S21T1\venv\libsite-packages\selenium\webdriver\remote\webdriver.py", line 394, in find_element_by_xpath return self.find_element(by=By.XPATH, value=xpath) File "C:\Users\Matthew\Desktop\project673\main\BUMETCS673S21T1\venv\libsite-packages\selenium\webdriver\remote\webdriver.py", line 976, in find_element return self.execute(Command.FIND_ELEMENT, (File "C:\Users\Matthew\Desktop\project673\main\BUMETCS673S21T1\venv\libsite-packages\selenium\webdriver\remote\webdriver.py", line 321, in execute self.error_handler.check_response(response) File "C:\Users\Matthew\Desktop\project673\main\BUMETCS673S21T1\venv\libsite-packages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response raise exception_class(message, screen, stacktrace) selenium.common.exceptions.NoSuchElementException: Message: no such element: Unable to locate element: {"method":"xpath","selector":"/html[@id='events']/div/div/div/div/div/div[2]/a/div/div/h3"} (Session info: chrome=90.0.4430.93)</p>		
test_TC_09	Error	View
test_TC_10	Error	View
test_TC_11	Error	View
test_TC_12	Error	View

Total: 12, Pass: 7, Error: 5 -- Duration: 121.77 s

# System Testing: testing webpage UI

## Unittest Results

Start Time: 2021-04-26 12:56:56

Duration: 34.71 s

Summary: Total: 3, Pass: 3

__main__.WebpageTesting	Status
test_dashboard	Pass
test_event	Pass
test_homepage	Pass

Total: 3, Pass: 3 – Duration: 34.71 s

```
def test_homepage(self):
    self.driver.get('http://localhost:3000/')
    time.sleep(1)
    # log in button
    log_in = self.driver.find_element_by_xpath('//*[@id="navigation-website"]/nav/div/a[1]').text
    assert log_in == 'Log In'
    # sign up button
    sign_up = self.driver.find_element_by_xpath('//*[@id="navigation-website"]/nav/div/a[2]').text
    assert sign_up == 'Sign Up'
    # start planning button
    start_planning = self.driver.find_element_by_xpath('//*[@id="product"]/div/div[1]/div/div[2]/a').text
    assert start_planning == 'Starting Planning'
    # plan.ly homepage button
    homepage = self.driver.find_element_by_xpath('//*[@id="navigation-website"]/nav/span/a').text
    assert homepage == 'Plan.ly'
```

# Testing Metrics



Metric ID	Metric Description	End of Iteration 0	End of Iteration 1	End of Iteration 2	End of Iteration 3
M-1	Testing coverage	0	4	13	27
M-2	Unit tests written	0	0	7	7
M-3	Unit tests passing	0	0	7	7
M-4	System test written	3	4	6	20
M-5	System test simulated	0	4	6	17
M-6	System tests passing	0	4	6	17



# Security

---



# Security

- Cross Site Request Forgery (CSRF)
  - CSRF tokens are unique, secret, and randomly generated values that are generated by a server-side application and sent to the client.
  - When a request is made from the client, the server checks to see if the request includes the expected CSRF token, and if it does, validates it.
- Django
  - Django comes with a user authentication system. It handles user accounts, groups, permissions and cookie-based user sessions.
  - The auth system consists of:
    - ❑ Users,Permissions: Binary (yes/no) flags designating whether a user may perform a certain task,Groups: A generic way of applying labels and permissions to more than one user,A configurable password hashing system,Forms and view tools for logging in users, or restricting content,A pluggable backend system
- JSON Web Token
  - open standard that defines a securely transmitting information between parties as a JSON object, information can be verified using digital sign.
  - JSON Web Tokens consist of three parts separated by dots (.), which are:
    - ❑ Header,Payload,Signature



# Testing: Non Functional Requirements

- Security
- User Interface
- Complexity
- Scalability

# Deployment

---

# CI/CD Pipeline

- We setup two Github Actions workflows to create a small CI/CD process
- Each pull request and commit was automatically linted by a combination of linters
- Every pull request is automatically assigned three reviewers to go over the code and make sure it passes its linting checks
- When a pull request is approved, it is merged with development, which is then pushed to Heroku
- New changes are pulled from development into feature branches

```
170 2021-04-04 15:55:52 [INFO] -----
171 2021-04-04 15:55:52 [INFO] File[public/index.html]
172 2021-04-04 15:55:52 [INFO] - File[index.html] was linted with [trainlint] successfully
173 2021-04-04 15:55:52 [INFO] -----
174 2021-04-04 15:55:52 [INFO] -----
175 2021-04-04 15:55:52 [INFO] Linting [SHOAGUIST.PS] files...
176 2021-04-04 15:55:52 [INFO] -----
177 2021-04-04 15:55:52 [INFO] -----
178 2021-04-04 15:55:52 [INFO] File[src/App.js]
179 2021-04-04 15:55:54 [INFO] - File[App.js] was linted with [eslint] successfully
180 2021-04-04 15:55:54 [INFO] -----
181 2021-04-04 15:55:54 [INFO] File[src/component/authentication/login-form/login-form.js]
182 2021-04-04 15:55:55 [ERROR] Found errors in [eslint] linting
183 2021-04-04 15:55:55 [ERROR] Error code: 1. Command output:
184 -----
185 170 /github/workspace/src/component/authentication/login-form/login-form.js
186 171 // error Definition for rule 'jsx-ally/label-has-associated-control' was not found. [jsx-ally/label-has-associated-control]
187 -----
188 X 1 problem (1 error, 0 warnings)
189 -----
190 2021-04-04 15:55:55 [INFO] -----
191 2021-04-04 15:55:55 [INFO] File[src/component/authentication/confirm-form/confirm-form.js]
192 2021-04-04 15:55:57 [INFO] - File[confirm-form.js] was linted with [eslint] successfully
193 2021-04-04 15:55:57 [INFO] -----
194 2021-04-04 15:55:57 [INFO] File[src/component/events/event/event.js]
195 2021-04-04 15:55:58 [INFO] - File[event.js] was linted with [eslint] successfully
196 2021-04-04 15:55:58 [INFO] -----
197 2021-04-04 15:55:58 [INFO] File[src/component/events/event-form/event-form.js]
198 2021-04-04 15:55:59 [INFO] - File[event-form.js] was linted with [eslint] successfully
199 2021-04-04 15:55:59 [INFO] -----
200 2021-04-04 15:55:59 [INFO] File[src/component/events/invite-list/invite-list.js]
201 2021-04-04 15:56:00 [INFO] - File[invite-list.js] was linted with [eslint] successfully
202 2021-04-04 15:56:00 [INFO] -----
203 2021-04-04 15:56:00 [INFO] File[src/component/events/rsvp-list/rsvp-list.js]
204 2021-04-04 15:56:00 [INFO] - File[rsvp-list.js] was linted with [eslint] successfully
205 2021-04-04 15:56:00 [INFO] -----
206 2021-04-04 15:56:00 [INFO] File[src/component/layout/dashboard/dashboard.js]
207 2021-04-04 15:56:00 [INFO] - File[dashboard.js] was linted with [eslint] successfully
208 2021-04-04 15:56:00 [INFO] -----
209 2021-04-04 15:56:00 [INFO] File[src/component/layout/navigation/navigation.js]
210 2021-04-04 15:56:00 [INFO] - File[nav.js] was linted with [eslint] successfully
211 2021-04-04 15:56:00 [INFO] -----
212 2021-04-04 15:56:00 [INFO] -----
213 2021-04-04 15:56:00 [ERROR] Found errors in [eslint] linting
214 2021-04-04 15:56:00 [ERROR] Error code: 1. Command output:
215 -----
216 210 /github/workspace/src/component/layout/navigation/navigation.js
217 211 // error Definition for rule 'jsx-ally/no-static-element-interactions' was not found. [jsx-ally/no-static-element-interactions]
218 212 // error Definition for rule 'jsx-ally/label-has-associated-control' was not found. [jsx-ally/label-has-associated-control]
219 -----
220 X 2 problems (2 errors, 0 warnings)
221 -----
222 2021-04-04 15:56:00 [INFO] -----
223 2021-04-04 15:56:00 [INFO] -----
224 2021-04-04 15:56:00 [INFO] Linting [JSDOM] files...
225 2021-04-04 15:56:00 [INFO] -----
```

```

Lint Code Base
failed yesterday in 2m 21s

▼ Lint Code Base
192 2021-04-27 01:10:30 [INFO] ~ file[App.js] was linted with [jscpd] successfully
193 -----
194 2021-04-27 01:10:30 [INFO] file[src/components/events/event/event.js]
195 2021-04-27 01:10:31 [INFO] ~ file[event.js] was linted with [jscpd] successfully
196 -----
197 2021-04-27 01:10:31 [INFO] file[src/components/events/eventform/eventform.js]
198 2021-04-27 01:10:32 [INFO] ~ file[eventform.js] was linted with [jscpd] successfully
199 -----
200 2021-04-27 01:10:32 [INFO] file[src/components/events/tasks/tasks/tasks.js]
201 2021-04-27 01:10:33 [ERROR] Found error in [jscpd] linting
202 2021-04-27 01:10:33 [ERROR] Error code: 1. Command output:
203 -----
204 Clone found {source:
205   - github/workspace/src/components/events/tasks/tasks.js [37:5 - 52:4] (5 lines, 60 tokens)
206   /github/workspace/src/components/events/tasks/tasks.js [21:7 - 33:6]
207 }
208 Clone found {source:
209   - github/workspace/src/components/events/tasks/tasks.js [37:5 - 52:4] (5 lines, 60 tokens)
210   /github/workspace/src/components/events/tasks/tasks.js [21:7 - 33:6]
211 }
212 39 | 21 | // GET, POST, PUT, DELETE, etc.
213 39 | 22 | headers: {
214 39 | 23 |   Authorization: `JWT ${localStorage.getItem('token')}`,
215 48 | 24 |   'X-CSRF-Token': csrfToken,
216 41 | 25 |   'Content-Type': 'application/json',
217 42 | 26 | },
218 43 | 27 | credentials: 'same-origin',
219 44 | 28 | body: JSON.stringify(details), // body data type must match "Content-Type" header
220 45 | 29 | });
221 45 | 30 | return response.json(); // parses JSON response into native JavaScript objects
222 47 | 31 | }
223 48 | 32 | }
224 49 | 33 | /**
225 50 | 34 |  * Class representing the Dashboard once user logs in.
226 51 | 35 |  * @extends Component
227 52 | 36 |  */
228 -----
229 Found 3 clones.
230 ERROR: jscpd found too many duplicates (4.50%) over threshold (0%)
231 Error: jscpd found too many duplicates (4.50%) over threshold (0%)
232 at ThresholdReporter.report (node_modules/@jscpd/indexer/dist/reporters/threshold.js:12:19)
233 at EventEmitter@jscpd/indexer/dist/in-files-detector.js:81:26
234 at Array.forEach ()
235 at node_modules/@jscpd/indexer/dist/in-files-detector.js:81:28
236 at processTicksAndRejections (node:internal/process/task_queues:94:5)
237 -----
238 2021-04-27 01:10:33 [INFO] ~ file[src/components/invoice/invoice.js]
239 2021-04-27 01:10:34 [INFO] ~ file[invoice.js] was linted with [jscpd] successfully
240 -----
241 2021-04-27 01:10:34 [INFO] file[src/components/layout/dashboard/dashboard.js]
242 2021-04-27 01:10:35 [INFO] ~ file[dashboard.js] was linted with [jscpd] successfully
243 -----
244 2021-04-27 01:10:35 [INFO] file[src/components/list/list.js]
245 2021-04-27 01:10:36 [INFO] ~ file[list.js] was linted with [jscpd] successfully
246 -----
247 2021-04-27 01:10:36 [INFO] file[src/index.css]
248 2021-04-27 01:10:36 [INFO] ~ file[styles.css] was linted with [jscpd] successfully

```

Lint Code Base

succeeded 21 hours ago in 2m 5s

▼

Limit Code Base

51s

215 2021-04-27 05:06:10 [INFO] File [backend/wr1.py] - File [backend/wr1.py] was linted with [pylint] successfully

216 2021-04-27 05:06:10 [INFO] -----

217 2021-04-27 05:06:10 [INFO] File [core/models.py] - File [models.py] was linted with [pylint] successfully

218 2021-04-27 05:06:10 [INFO] -----

219 2021-04-27 05:06:10 [INFO] File [core/serializer.py] - File [serializer.py] was linted with [pylint] successfully

220 2021-04-27 05:06:10 [INFO] -----

221 2021-04-27 05:06:10 [INFO] File [core/wr1.py] - File [wr1.py] was linted with [pylint] successfully

222 2021-04-27 05:06:10 [INFO] -----

223 2021-04-27 05:06:10 [INFO] File [core/view.py] - File [view.py] was linted with [pylint] successfully

224 2021-04-27 05:06:10 [INFO] -----

225 2021-04-27 05:06:10 [INFO] -----

226 2021-04-27 05:06:10 [INFO] -----

227 2021-04-27 05:06:10 [INFO] -----

228 2021-04-27 05:06:10 [INFO] -----

229 2021-04-27 05:06:10 [INFO] -----

230 2021-04-27 05:06:10 [INFO] -----

231 2021-04-27 05:06:10 [INFO] -----

232 2021-04-27 05:06:10 [INFO] -----

233 2021-04-27 05:06:10 [INFO] -----

234 2021-04-27 05:06:10 [INFO] -----

235 2021-04-27 05:06:10 [INFO] -----

236 2021-04-27 05:06:10 [INFO] -----

237 2021-04-27 05:06:10 [INFO] File [backend/settings.py] - File [settings.py] was linted with [mypy] successfully

238 2021-04-27 05:06:10 [INFO] -----

239 2021-04-27 05:06:10 [INFO] File [backend/wr1.py] - File [wr1.py] was linted with [mypy] successfully

240 2021-04-27 05:06:10 [INFO] -----

241 2021-04-27 05:06:10 [INFO] -----

242 2021-04-27 05:06:10 [INFO] File [core/models.py] - File [models.py] was linted with [mypy] successfully

243 2021-04-27 05:06:10 [INFO] -----

244 2021-04-27 05:06:10 [INFO] File [core/serializer.py] - File [serializer.py] was linted with [mypy] successfully

245 2021-04-27 05:06:10 [INFO] -----

246 2021-04-27 05:06:10 [INFO] File [core/wr1.py] - File [wr1.py] was linted with [mypy] successfully

247 2021-04-27 05:06:10 [INFO] -----

248 2021-04-27 05:06:10 [INFO] File [core/view.py] - File [view.py] was linted with [mypy] successfully

249 2021-04-27 05:06:10 [INFO] -----

250 2021-04-27 05:06:10 [INFO] -----

251 2021-04-27 05:06:10 [INFO] -----

252 2021-04-27 05:06:10 [INFO] -----

253 2021-04-27 05:06:10 [INFO] -----

254 2021-04-27 05:06:10 [INFO] -----

255 2021-04-27 05:06:10 [INFO] -----

256 2021-04-27 05:06:10 [INFO] -----

257 2021-04-27 05:06:10 [INFO] -----

258 2021-04-27 05:06:10 [INFO] -----

259 2021-04-27 05:06:10 [INFO] -----

260 2021-04-27 05:06:10 [INFO] -----

261 2021-04-27 05:06:10 [INFO] -----

262 2021-04-27 05:06:10 [INFO] -----

263 2021-04-27 05:06:10 [INFO] -----

264 2021-04-27 05:06:10 [INFO] -----

265 2021-04-27 05:06:29 [NOTICE] All file(s) linted successfully with no errors detected

266 2021-04-27 05:06:29 [INFO] -----

>

Post Checkout Code

8s

>

Complete job

8s



# Heroku Deployment

main/release branch - plan-ly.herokuapp.com

Development - development.planly.herokuapp.com

Heroku uses two buildpacks

-heroku/nodejs

-heroku/python

Each buildpack runs the build process for its respective framework and installs dependencies and modules into Heroku so that the app can run

```
-----> Building on the Heroku-20 stack
-----> Using Buildpacks:
   1. heroku/nodejs
   2. heroku/python
-----> Node.js app detected

-----> Creating runtime environment

      npm_config_log_level=error
      NODE_VERSION=false
      NODE_ENV=production
      NODE_MODULES_CACHE=true

-----> Installing binaries
   engines.node (package.json): 14.6.0...
   engines.npm (package.json):  6.14.9

   Resolving node version 14.6.0...
   Downloading and installing node 14.6.0...
   Bootstrapping npm 6.14.9 (replacing 6.14.6)...
   npm 6.14.9 installed

-----> Restoring cache
   - node_modules

-----> Installing dependencies
   Installing node modules

   > core-js@2.6.12 postinstall /tmp/build_4b7a20a3/node_modules/core-js
   > node -e "try(require('./postinstall'))catch(e){}"

   > ejs@2.7.4 postinstall /tmp/build_4b7a20a3/node_modules/ejs
   > node ./postinstall.js

   > fsevents@1.2.13 install /tmp/build_4b7a20a3/node_modules/webpack-dev-server/node_modules/fsevents
   > node install.js

   Skipping 'fsevents' build as platform linux is not supported

   > fsevents@1.2.13 install /tmp/build_4b7a20a3/node_modules/watchpack-chokidar2/node_modules/fsevents
   > node install.js

   Skipping 'fsevents' build as platform linux is not supported

   > core-js@3.9.1 postinstall /tmp/build_4b7a20a3/node_modules/react-app-polyfill/node_modules/core-js
   > node -e "try(require('./postinstall'))catch(e){}"
```

# Demo

---



---

Thank you

Questions?

