

# PlanningJam

---

Iteration 2



BU MET 673. Fall 2025 OL. Team 3

# Team

---

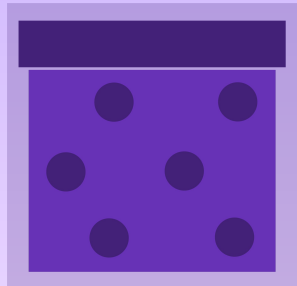
**David Metraux - Team Leader**

**Haolin Yang - Design and Implementation Leader**

**Donjay Barit - Configuration Leader**

**Ashley Sachdeva - Requirements Leader**

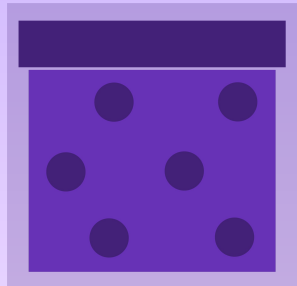
**Jason Lee - QA Leader**



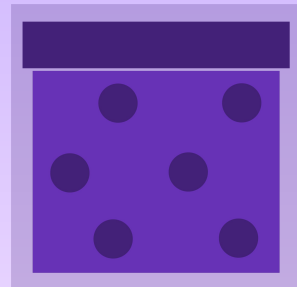
# Table of Contents

---

- Summary
- STD Document
- Future Plans

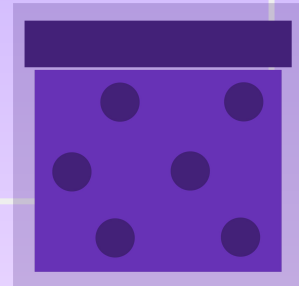


# Iteration 2 Summary



# Iteration 2 Summary

- Implemented in Frontend & Backend:
  - **Adding, Editing, Deleting** Plans
  - **Sending, Accepting, Deleting** Friend Requests
  - **RSVP** to Plans - only implemented in backend
- Set up testing structure for both frontend and backend
- Updated Postman environments and API Documentation



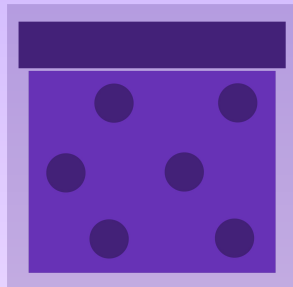
# Plans Feature

---



## Add / Edit / Delete

- **Secure design:** JWT authentication + owner-only edit/delete prevents unauthorized access
- **Smart storage:** Mongo-backed with auto-populated `created_by` field tracks ownership
- **CRUD flow:** Create → Validate → Store → Return ID (clean REST semantics)
- **Authorization:** Can't edit someone else's plans (403 Forbidden)
- **Error handling:** Handles validation errors, missing plans, auth failures with proper HTTP codes



# Friends Feature

---

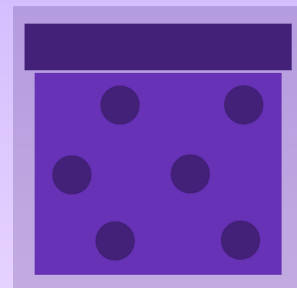
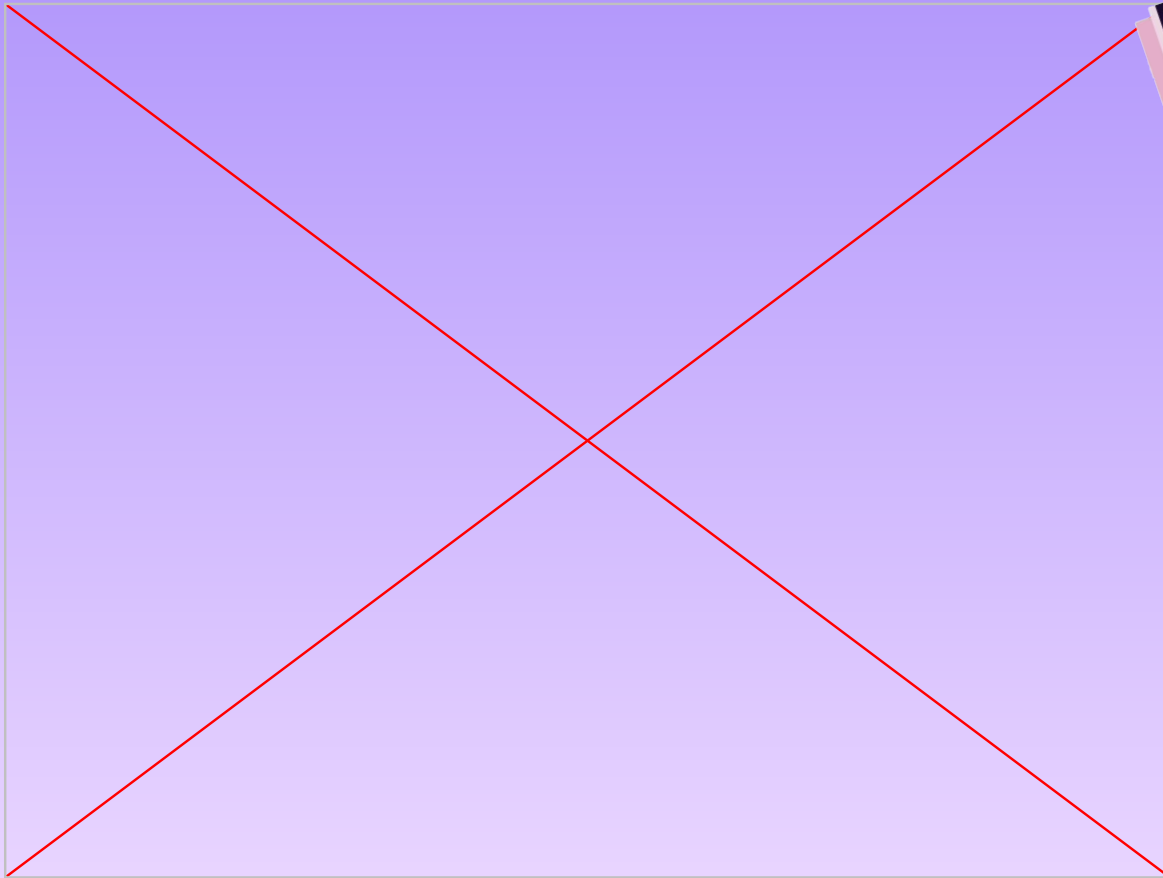
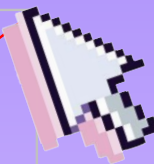
## Send / Accept / Delete

- **Duplicate prevention:** No spam requests or confusing double-pending scenarios
- **Auto-accept:** When  $A \rightarrow B$  and  $B \rightarrow A$  both happen, system merges into single accepted friendship
- **Flexible removal:** Delete by friend record ID with user ID fallback for edge cases



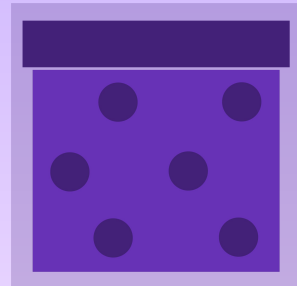
# Demo

---

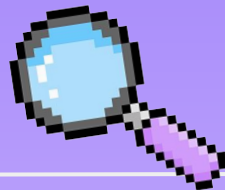




# STD Document



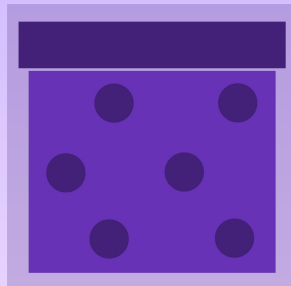
# Manual test coverage



## Added manual test cases for E2E testing between the frontend and backend for new features

### Test Case 6: **Add Plan**

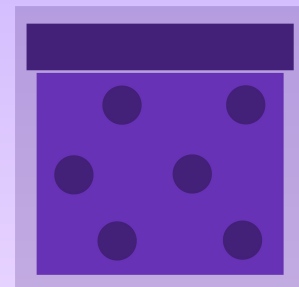
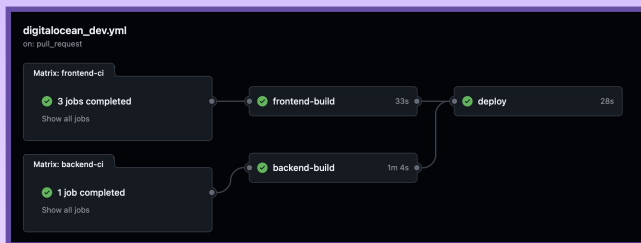
- Test items: Add new plan functionality
- Test priority: High
- Dependencies (to other test case/requirement if any): User must be logged in
- Preconditions (if any): Valid authenticated session, user on `/plans/add`
- Input data: plan title, description, location name, address, city, state, zipcode, start and end time
- Test steps:
  - Click "Add Plan" button on home page
  - Fill in only some of the data
  - Click "Save"
  - Ensure error message shows to fill out other input data
  - Click cancel button
  - Ensure it redirects to home page
  - Fill rest of input data
  - Click Save button
- Postconditions:
  - New plan is created and visible on the home page
- Expected output: The new plan appears immediately in the plans list with correct information
- Actual output: N/A
- Pass or Fail: N/A
- Bug id/link: N/A
- Additional notes: UI refreshes automatically after adding a plan; no manual reload needed.



# Automated test coverage



- Added automated test cases for specific component rendering on the frontend.
  - Used BDD and TDD approach to develop tests
  - Used Vitest and react testing library
  - Generated test coverage report
  - Tested error cases
  - Github Actions & DigitalOcean



# Testing Metrics



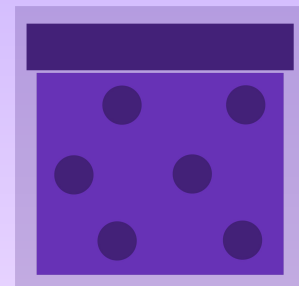
- Frontend 70% coverage on statements, branches, lines
- Frontend 60% coverage on functions

```
Test Files 7 passed (7)
Tests 29 passed (29)
Start at 20:32:23
Duration 3.96s (transform 528ms, setup 1.88s, collect 5.69s, tests 3.31s, environment 5.22s, prepare 517ms)
```

% Coverage report from v8

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	71.88	70.22	59.37	71.88	
src	79.78	100	100	79.78	
App.jsx	100	100	100	100	
main.jsx	0	100	100	0	13-32
src/auth	80.16	75.43	62.5	80.16	
AuthContext.tsx	69.62	71.42	66.66	69.62	43-44,60-64,97-104,107-112,114-117,132-137,153-156,171-177
AuthProtectedRoute.tsx	40	100	0	40	20-28
Login.tsx	80.95	40	25	80.95	28-29,33-44,84-86
Logout.tsx	30	100	0	30	15-23
Signup.tsx	98.30	87.5	100	98.30	68-69
src/auth/endpoints	1.61	100	0	1.61	
auth.ts	1.61	100	0	1.61	14-87
src/components	97.5	73.33	80	97.5	
Banner.jsx	100	100	100	100	
Home.tsx	94.25	69.23	66.66	94.25	37-39,49-50
Logo.jsx	100	100	100	100	
src/plans	97.17	64.28	70.83	97.17	
AddPlanForm.tsx	95.31	48.14	53.84	95.31	105-108,111-112,228-230
AddPlanPage.tsx	97.82	81.25	100	97.82	34
PlanCard.tsx	100	72.72	83.33	100	48,73,100
PlansHeader.tsx	100	100	100	100	
src/plans/endpoints	4.19	100	0	4.19	
addPlan.ts	3.44	100	0	3.44	14-57
deletePlan.ts	4.16	100	0	4.16	14-38
editPlan.ts	3.33	100	0	3.33	14-57
getPlan.ts	5.55	100	0	5.55	16-55
getPlanById.ts	4.16	100	0	4.16	30-55
src/styles	0	0	0	0	
theme.js	0	0	0	0	1-35
src/users/endpoints	3.84	100	0	3.84	
getUserById.ts	3.84	100	0	3.84	15-44

frontend git:(PJ-47) x

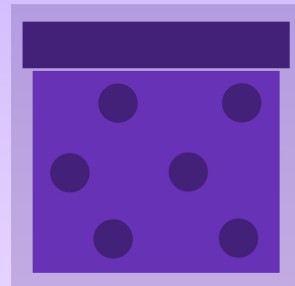


# Testing Metrics



- Backend 80% coverage on statements, branches, lines

Name	Stmts	Miss	Cover	Missing
__init__.py	0	0	100%	
api/_init_.py	0	0	100%	
api/admin.py	1	0	100%	
api/apps.py	4	0	100%	
api/management/_init_.py	0	0	100%	
api/migrations/0001_initial.py	9	0	100%	
api/migrations/0002_remove_friend_api_friend_sender_dfedbd_idx_and_more.py	4	0	100%	
api/migrations/0003_remove_friend_unique_sender_receiver_and_more.py	6	0	100%	
api/migrations/_init_.py	0	0	100%	
api/models/_init_.py	2	0	100%	
api/models/friends_models.py	21	5	76%	61, 64-65, 68-69
api/serializers/auth_serializer.py	32	10	69%	25-27, 30-32, 35-38
api/serializers/plans_serializer.py	15	0	100%	
api/tests/_init_.py	0	0	100%	
api/tests/plans_test.py	16	0	100%	
api/tests/test_friends.py	58	0	100%	
api/urls.py	5	0	100%	
api/utills/mongo.py	23	10	57%	50-61, 65
api/views/_init_.py	0	0	100%	
api/views/Friends.py	83	19	77%	37, 44, 76, 152, 157-183
api/views/plans.py	78	34	56%	54-62, 69-75, 82-93, 100-116, 123-133
api/views/users.py	46	21	54%	39-52, 63-65, 83-101, 113-128
app/_init_.py	0	0	100%	
app/apps.py	9	0	100%	
app/settings.py	29	0	100%	
app/urls.py	3	0	100%	
conftest.py	20	0	100%	
mongo_migrations/_init_.py	0	0	100%	
mongo_migrations/admin/0001_initial.py	10	0	100%	
mongo_migrations/admin/_init_.py	0	0	100%	
mongo_migrations/auth/0001_initial.py	10	0	100%	
mongo_migrations/auth/_init_.py	0	0	100%	
mongo_migrations/contenttypes/0001_initial.py	7	0	100%	
mongo_migrations/contenttypes/_init_.py	0	0	100%	
TOTAL	491	99	80%	



# Testing Metrics

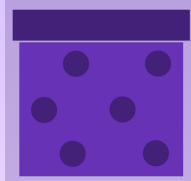


- LOC = 1,416,730
- Average Cyclomatic Complexity : A (3.08)

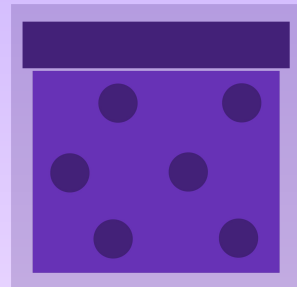
```
backend/api/models.py
C 19:0 Friend - A (2)
M 54:4 Friend._str - A (1)
M 57:4 Friend.accept - A (1)
M 61:4 Friend.reject - A (1)
backend/api/apps.py
C 4:0 AppConfig - A (1)
backend/api/migrations/0002_remove_friend_api_friend_sender_dfedbd_idx_and_more.py
C 6:0 Migration - A (1)
backend/api/migrations/0003_remove_friend_unique_sender_receiver_and_more.py
C 8:0 Migration - A (1)
backend/api/migrations/0001_initial.py
C 10:0 Migration - A (1)
backend/api/test/test_friends.py
F 70:0 test_list_friends - B (9)
F 21:0 test_send_friend_request - A (4)
F 35:0 test_reciprocal_request_auto_accept - A (4)
F 54:0 test_response_to_friend_request - A (4)
F 97:0 test_remove_friend - A (3)
backend/api/test/rsvp_test.py
M 29:4 TestRSVPAPI.test_create_rsvp_success - B (7)
M 86:4 TestRSVPAPI.test_get_rsvp_by_plan_id_success - B (7)
M 128:4 TestRSVPAPI.test_get_rsvp_by_user_id_success - B (7)
M 388:4 TestRSVPAPI.test_rsvp_data_integrity - A (5)
C 17:0 TestRSVPAPI - A (4)
M 53:4 TestRSVPAPI.test_create_rsvp_missing_plan_id - A (3)
M 116:4 TestRSVPAPI.test_get_rsvp_by_plan_id_not_found - A (3)
M 158:4 TestRSVPAPI.test_get_rsvp_by_user_id_not_found - A (3)
M 170:4 TestRSVPAPI.test_delete_rsvp_by_id_success - A (3)
M 185:4 TestRSVPAPI.test_delete_rsvp_by_id_not_found - A (3)
M 199:4 TestRSVPAPI.test_delete_rsvp_by_id_invalid_id - A (3)
M 280:4 TestRSVPAPI.test_delete_rsvp_by_plan_id_success - A (3)
M 222:4 TestRSVPAPI.test_delete_rsvp_by_plan_id_not_found - A (3)
M 235:4 TestRSVPAPI.test_delete_rsvp_by_plan_id_invalid_id - A (3)
M 257:4 TestRSVPAPI.test_multiple_rsups_same_plan - A (3)
M 284:4 TestRSVPAPI.test_duplicate_rsvp_prevention - A (3)
M 65:4 TestRSVPAPI.test_create_rsvp_invalid_plan_id - A (2)
M 243:4 TestRSVPAPI.test_rsvp_authentication_required - A (2)
M 19:4 TestRSVPAPI.setup_method - A (1)
backend/api/test/plans_test.py
C 15:0 TestPlansAPI - A (5)
M 16:4 TestPlansAPI.test_create_plan - A (4)
backend/api/management/commands/seed_dev_users.py
C 5:0 Command - A (5)
M 8:4 Command.handle - A (4)
backend/api/utils/mongo.py
F 31:0 get_collection - B (6)
F 14:0 unwrap_db - A (3)
```

```
backend/api/models/friends_models.py
C 20:0 Friend - A (2)
M 59:0 Friend._str - A (1)
M 63:4 Friend.accept - A (1)
M 67:4 Friend.reject - A (1)
backend/api/serializers/rsvp_serializer.py
C 8:0 RSVPSerializer - A (1)
backend/api/serializers/friends_serializer.py
C 20:0 FriendSerializer - A (2)
M 30:4 FriendSerializer.create - A (1)
M 33:4 FriendSerializer.update - A (1)
backend/api/serializers/fields.py
C 17:0 ObjectIdField - A (3)
M 18:4 ObjectIdField.to_representation - A (2)
M 24:4 ObjectIdField.to_internal_value - A (2)
backend/api/serializers/auth_serializer.py
C 13:0 UserRegistrationSerializer - A (3)
M 21:4 UserRegistrationSerializer.validate_email - A (2)
M 29:4 UserRegistrationSerializer.validate - A (2)
M 34:4 UserRegistrationSerializer.create - A (1)
C 40:0 UserSerializer - A (1)
backend/api/serializers/plans_serializer.py
C 7:0 LocationSerializer - A (1)
C 15:0 PlansSerializer - A (1)
backend/api/views/friends.py
F 140:0 remove_friend - B (10)
F 35:0 send_friend_request - B (6)
F 87:0 list_friends - A (5)
F 67:0 respond_to_friend_request - A (4)
backend/api/views/users.py
F 72:0 list_users - A (5)
F 107:0 get_user - A (5)
F 33:0 register_user - A (2)
F 58:0 get_user_profile - A (1)
backend/api/views/rsvp.py
F 25:0 create_rsvp - B (6)
F 68:0 get_rsvp_by_plan_id - A (4)
F 87:0 get_rsvp_by_user_id - A (4)
F 106:0 delete_rsvp_by_id - A (3)
F 122:0 delete_rsvp_by_plan_id - A (3)
backend/api/views/plans.py
F 28:0 create_plan - A (5)
F 81:0 get_plans_by_id - A (3)
F 99:0 update_plan - A (3)
F 122:0 delete_plan - A (3)
F 68:0 get_plans - A (2)
F 20:0 get_plans_collection - A (1)

75 blocks (classes, functions, methods) analyzed.
Average complexity: A (3.08)
```



# Future Plans



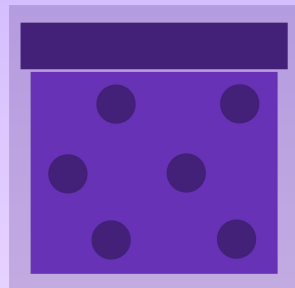
# Moving Forward

---



For the next iteration, we plan to:

- **Backend:** Add more functionality such as dismiss plans, profile management
- **Frontend:** Add corresponding UI components and functionality to RSVP and filter plans
- Add more security checks within code base
- Increase test coverage





**Thanks for Watching**

