

PlanningJam

Iteration 3



BU MET 673. Fall 2025 OL. Team 3

Team Roles ✨

Name	Role
David	Team Leader
Ashley	Requirement Leader
Donjay	Configuration and Security Leader
Haolin	Design and Implementation Leader
Jason	QA Leader

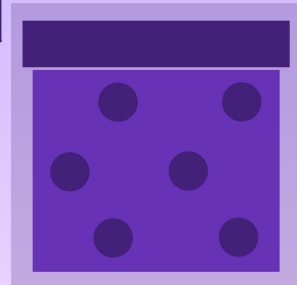
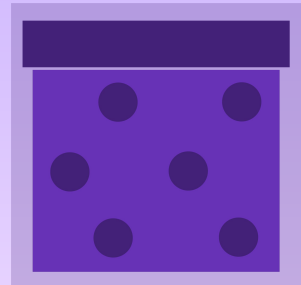
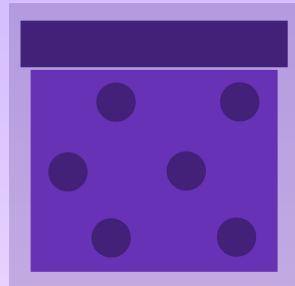


Table of Contents

- Iteration 3 Summary
- Individual Report
 - Project Design & Implementation
 - Management
 - System Infrastructure
 - Requirements and Design
 - Project QA



Iteration 3 Summary



RSVP Plan Functionality

Can RSVP to plans created by friends and see others that are going

Hockey game



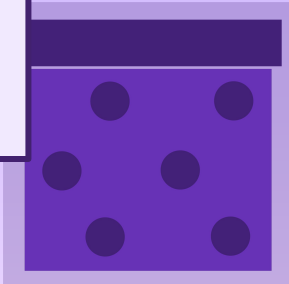
 Hosted By: [asach12345](#)

10/30/2025, 10:25:00 PM – 10/31/2025, 12:35:00 AM

bc hockey

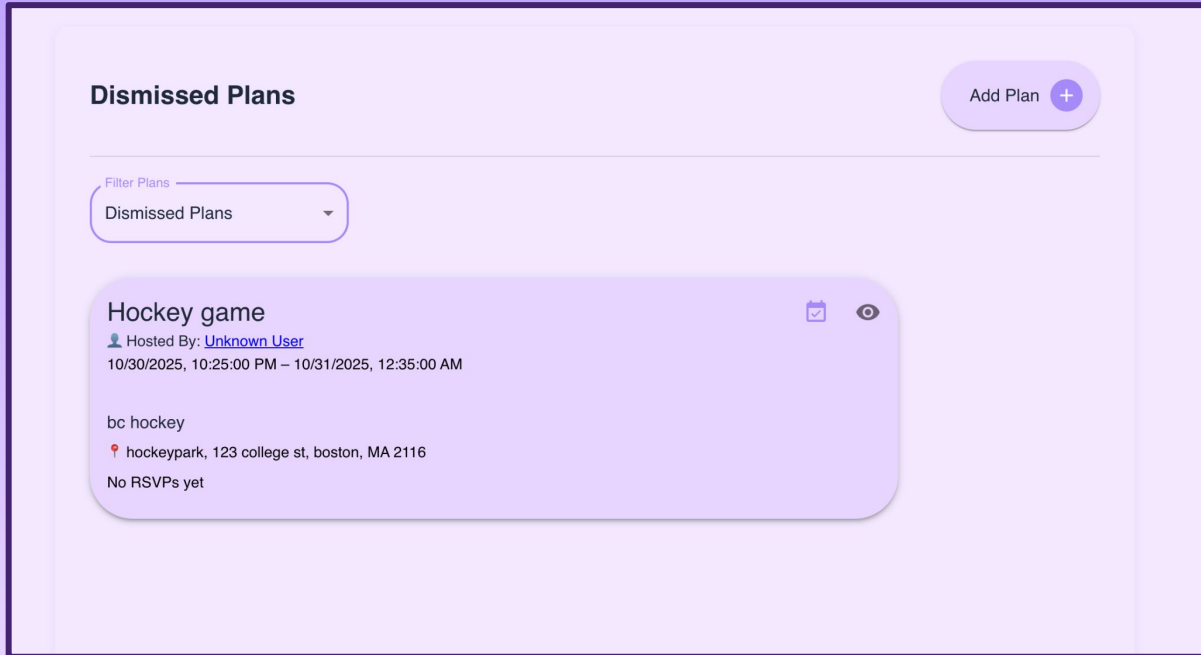
 hockeypark, 123 college st, boston, MA 2116

Going: [asach8888](#)



Dismiss Plan Functionality

Option to dismiss a plan you don't want to see on your main screen anymore. Can filter if you want to see plans you dismissed and undismiss them.



Frontend: Friends

Previous work:
Uses MUI now!

Friends

Current Friends:

[Test](#) REMOVE

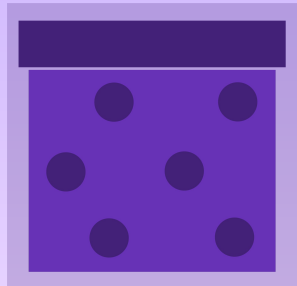
Add Friends:

[DavidMetraux](#) SEND

Received Friend Requests:

Sent Friend Requests:

[Testing](#)



Frontend: Banner

Desktop Widths:

PlanningJam 



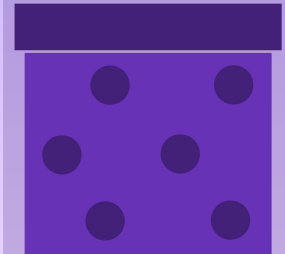
LOGOUT

Mobile
Widths:

PlanningJam

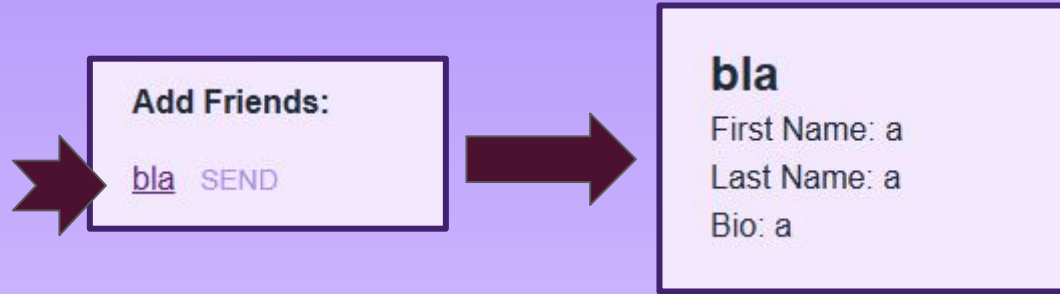


LOGOUT

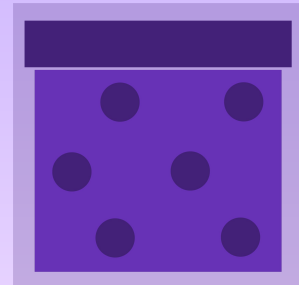
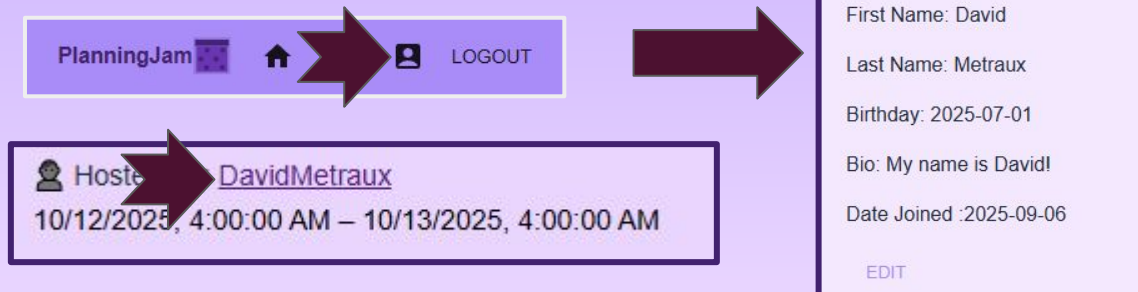


Frontend: User Bios and Profiles

If click on another user's name: `/users/68e961308207bba0fd61b5a5`



If click on your own name, or the profile icon in the banner `/profile`



Frontend: Editing Profiles

DavidMetraux

Email: metrauxdavid@gmail.com

First Name: David

Last Name: Metraux

Birthday: 2025-07-01

Bio: My name is David!

Date Joined :2025-09-06

EDIT



Edit Profile

First Name

David

Last Name

Metraux

Date of Birth

07/01/2025



Bio

My name is David!

UPDATE

CANCEL



DavidMetraux

Email: metrauxdavid@gmail.com

First Name: Dave

Last Name: Metro

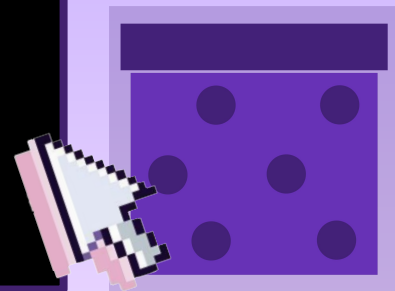
Birthday: 1997-12-26

Bio: My name is David! I love programming software.

Date Joined :2025-09-06

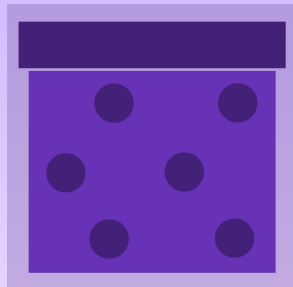
EDIT

Full Demo

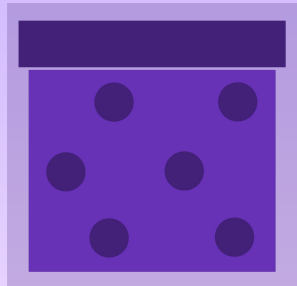
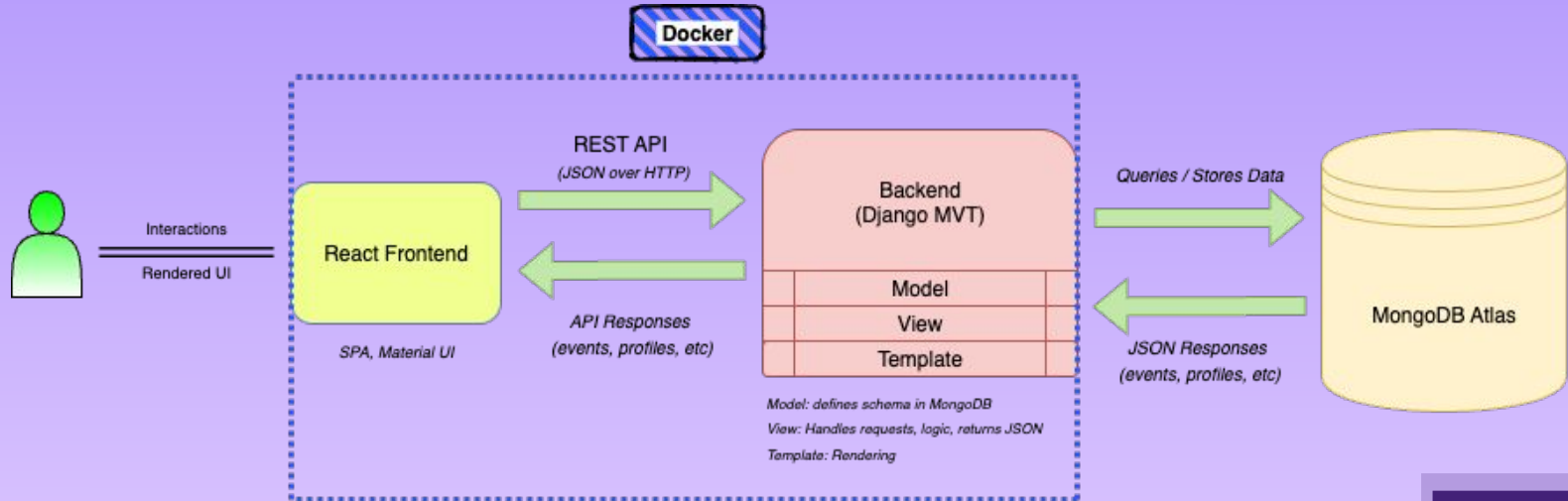


Project Design & Implementation

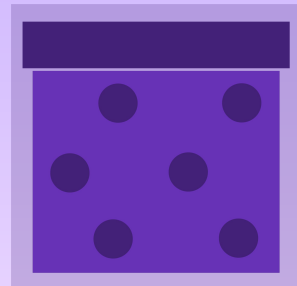
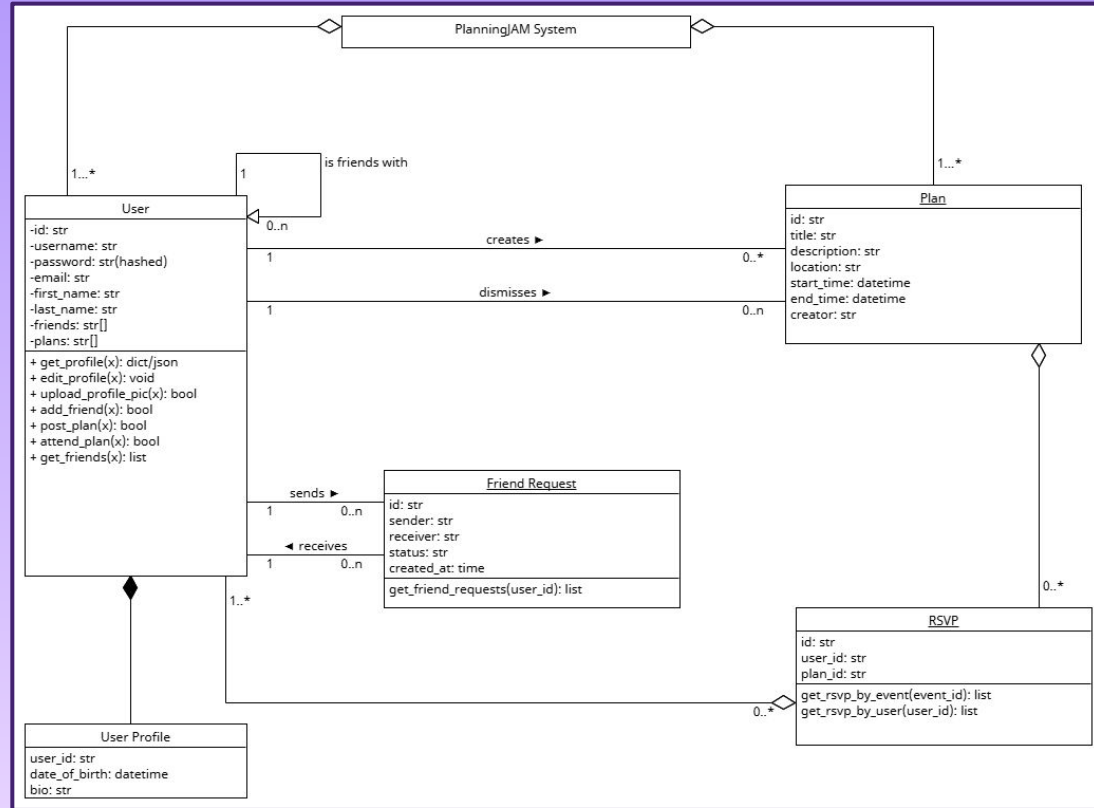
Haolin Yang, Design and Implementation Leader



System Architecture



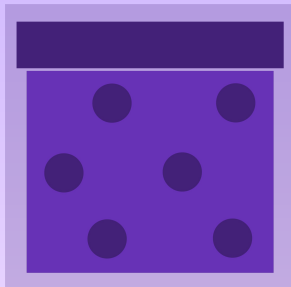
Class Diagram



Mistakes & Lessons Learned

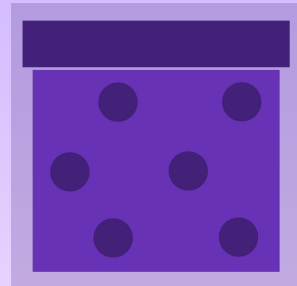
Good Design is Iterative!

Mistakes	What I learned
User class uses the default Django user, which made it almost impossible to incorporate additional fields neatly.	Plan model flexibility upfront — it's easier to simplify a flexible design later than to refactor a rigid one.
Some related features were not completed in order. It was way too late to change the implementation.	When working on the application, always try to see the full picture instead of feature by feature.
Tried to over-engineer features we couldn't finish.	Scope realistically and prioritize a stable core.
Didn't fully consider frontend needs when defining APIs.	Communicate earlier with frontend before finalizing backend schema.



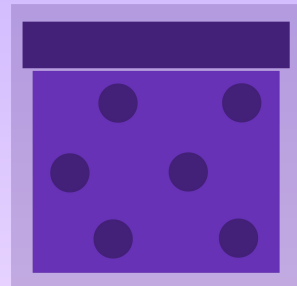
Management

David Metraux, Team Leader



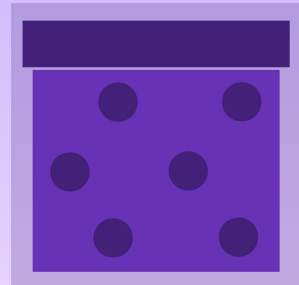
Management Strategies

- Frequent communication
- Delegation of tasks
- Rewatching lectures to gather pertinent information
- Stay up late to finish iterations with others
- General administrative work
 - Scheduling meetings
 - Writing agenda points
 - Progress reports
 - Timeline adjustments
 - Submitting work



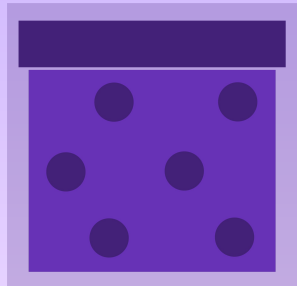
Risk Assessment/Issues

- A few times where team members had to consult others to understand software better
- Sometimes servers were down
- Both David and Ashley were unavailable for some time
 - Should have delegated more tasks during periods of unavailability
- Sometimes had to stay up late to finish iterations



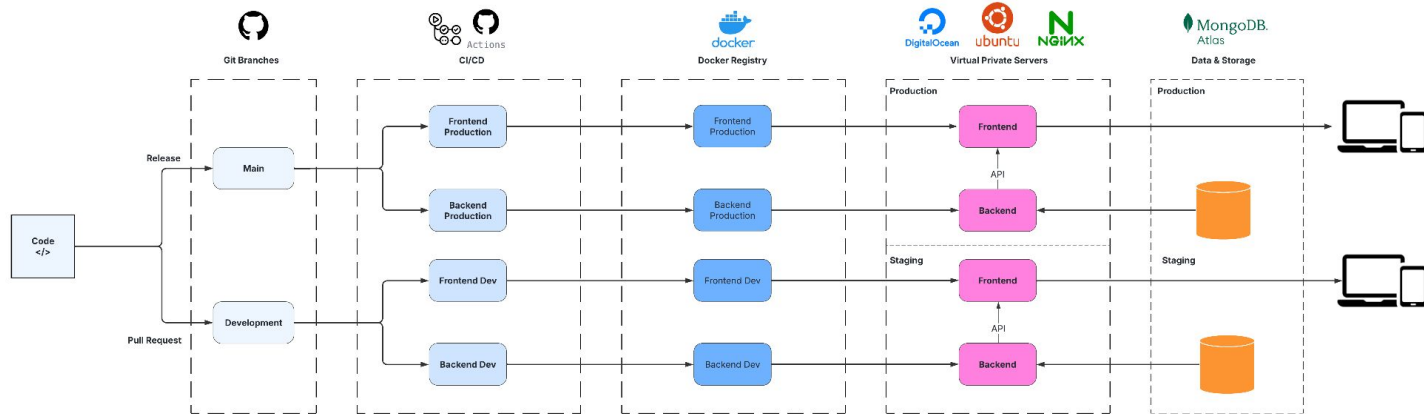
Deployment & System Infrastructure

Donjay Barit, Configuration and Security Leader



Deployment

Planning Jam CI/CD Pipeline



Matrix: frontend-ci

✓ 3 jobs completed

Show all jobs

✓ frontend-build

43s

✓ frontend-deploy

9s

Matrix: backend-ci

✓ 1 job completed

Show all jobs

✓ backend-build

57s

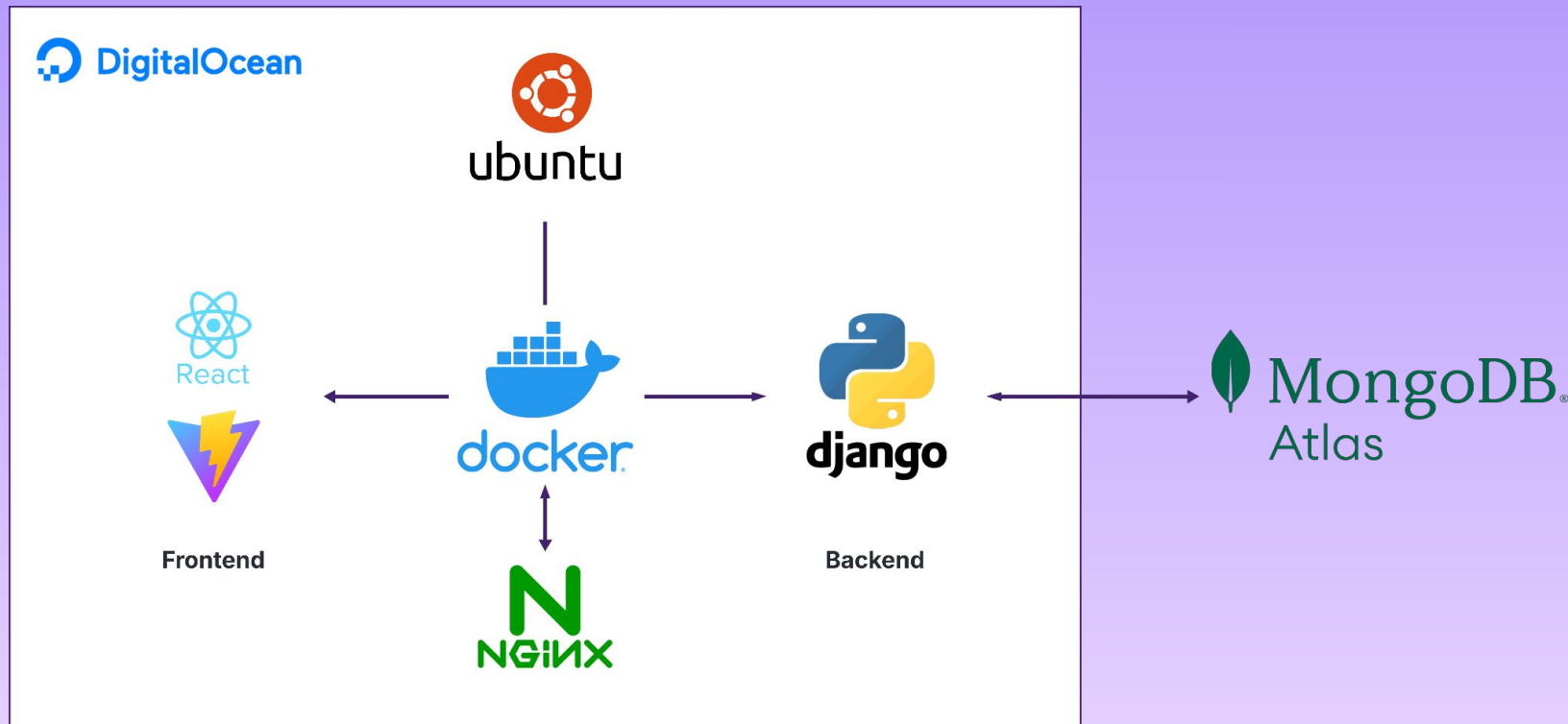
✓ backend-deploy

16s

IMAGES (7)

	Digest ?	Tags	Size ?
<input type="checkbox"/>	sha256:5d30f4211655	latest 2.1.1	24.11 MB
<input type="checkbox"/>	sha256:14ad1c50e16c	2.1	21.64 MB
<input type="checkbox"/>	sha256:63f25d081eaf	2	21.64 MB
<input type="checkbox"/>	sha256:602403544a31	1.1	21.56 MB

Server Stack



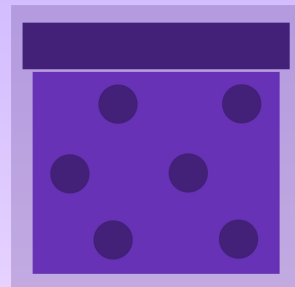
Database Design

Database: MongoDB

Collections:



- Users - stores user profiles and authentication related data
- Plans - stores plan informations
- RSVP - track relationships between user and plans
- Friends - friend relationship between users
- Dismissals - track relationship between user and dismissed plans



Security Design

Code Scan Security - Vulnerability and code exposure scanning

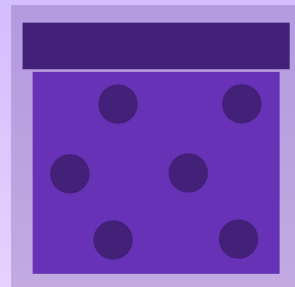
Application Security - JWT session tokens

Deployment Security - Securing secret keys and environment variables

Infrastructure Security - SSH key authentications (passwordless), firewalls

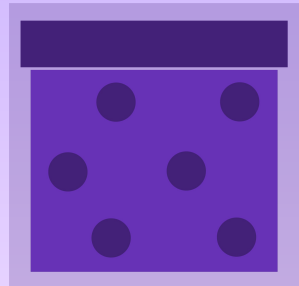
DNS Security - Proxied IPs, Strict SSL/TLS, DDoS defense

Database Security - Encrypted data, Whitelist IPs



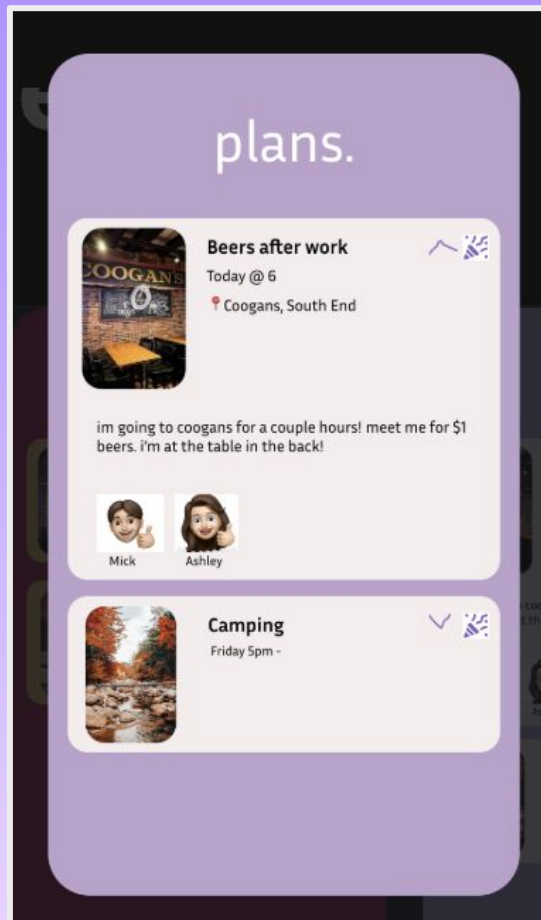
Requirements and Design

Ashley Sachdeva, Requirements Leader



Design

- Initial mock ups of the project
- Did not use exact UI but similar theming to integrate with material UI
- More mobile focused initial design - expanded to web



Requirements

- Created high level functional requirements
- Ensured we achieved most of them with the time and constraints
- Scoped out other levels of requirements (non-functional, desirable, optional)

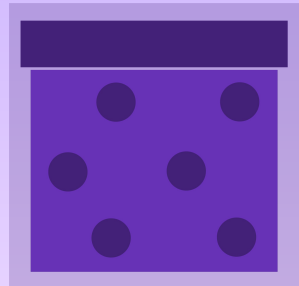
Functional Requirements

Essential Features

- **User Authentication**
 - **Description:** As a user, I want to log in and create an account, so that my plans and responses are secure and personalized.
 - **Estimated Person-Hours:** 5–10 hours
- **Post a Plan**
 - **Description:** As a user, I want to post a plan, so that all my friended users can see and respond to it.
 - **Estimated Person-Hours:** 5–10 hours
- **RSVP (Yes Response)**
 - **Description:** As a user, I want to respond to my friends' plans with Yes, so that I can indicate whether I will participate.
 - **Estimated Person-Hours:** 2-5 hours
- **Filter Plans by Type**
 - **Description:** As a user, I want to filter plans by type (e.g., food, sports, study), so that I can easily find plans I'm interested in.
 - **Estimated Person-Hours:** 1–3 hours
- **Friend Requests**
 - **Description:** As a user, I want to send and accept friend requests, so that I can connect with people I want to plan with.

Project Quality Assurance

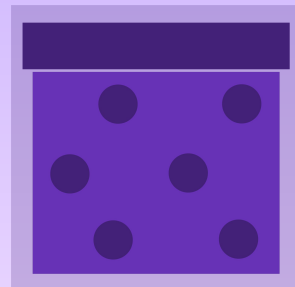
Jason Lee, QA



Automated Test Coverage



- Tools: Vitest + RTL (frontend), pytest + coverage.py (backend)
- Approach: TDD/BDD where feasible; negative tests (error cases) included
- Github Actions runs on every PR



Frontend Testing



- Used vitest to test frontend code
- Added component level testing
- Tested different branches, functions, and mocked backend responses
- Added manual testing steps for e2e tests

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.

Testing Metrics - Frontend



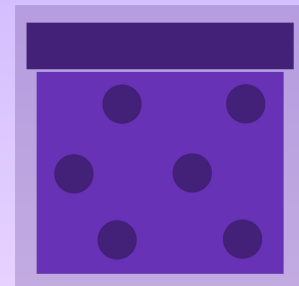
- 45-65% coverage on statements, branches, lines

```
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	28-28
Login.tsx	80.95	40	25	80.95	28-29, 33-44, 84-86
Logout.tsx	30	100	0	30	15-23
Signup.tsx	98.38	87.5	100	98.38	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

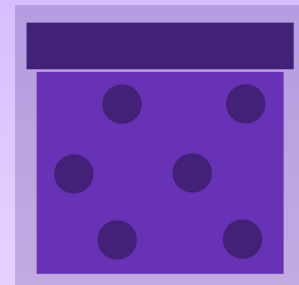


- 80% coverage on statements, branches, lines

```
===== tests coverage =====
coverage: platform darwin, python 3.13.7-final-0
=====
```

Name	Stats	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/management/commands/_init_.py	0	0	100%	
api/management/commands/reset_dev_users.py	18	18	0%	1-44
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/0004_userprofile.py	7	0	100%	
api/migrations/_init_.py	0	0	100%	
api/models.py	21	21	0%	15-63
api/models/_init_.py	3	0	100%	
api/models/friends_models.py	21	5	76%	61, 64-65, 68-69
api/models/user_models.py	12	1	92%	53
api/serializers/auth_serializer.py	58	0	100%	
api/serializers/plans_serializer.py	22	1	95%	38
api/serializers/rsvp_serializer.py	5	0	100%	
api/services/_init_.py	0	0	100%	
api/services/friends.py	10	0	100%	
api/services/plans.py	125	56	55%	23-24, 44, 50, 85-87, 90-92, 113, 139-140, 149-158, 173-220, 237, 240, 247, 251, 261-263, 267-272
api/tests/_init_.py	0	0	100%	
api/tests/plans_test.py	79	0	100%	
api/tests/rsvp_test.py	282	0	100%	
api/tests/test_friends.py	58	0	100%	
api/tests/test_plans_dismissal.py	125	0	100%	
api/tests/test_users.py	226	0	100%	
api/urls.py	5	0	100%	
api/urls/_init_.py	0	0	100%	
api/urls/getting.py	9	4	56%	12-15, 24
api/urls/mongo.py	35	11	69%	50-61, 65, 126
api/views/_init_.py	0	0	100%	
api/views/friends.py	13	15	77%	37, 44, 76, 152, 157-183
api/views/plans.py	100	9	91%	48-55, 89, 103-104, 110, 124-125
api/views/rsvp.py	83	7	92%	54-56, 71-72, 88-89
api/views/tokens.py	6	0	100%	
api/views/users.py	57	0	100%	
app/_init_.py	0	0	100%	
app/apps.py	9	0	100%	
app/asgi.py	4	4	0%	13-19
app/settings.py	29	0	100%	
app/urls.py	3	0	100%	
app/wsgi.py	4	4	0%	10-16
conf/test.py	20	0	100%	
manage.py	11	11	0%	3-22
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	1501	171	89%	

```
Coverage HTML written to dir htmlcov
===== 59 passed in 21.71s =====
```



Testing Metrics - LOC

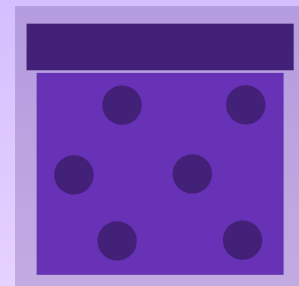


- LOC = 12,823 (excluding blanks)

Language	files	blank	comment	code
JSON	4	0	0	5659
TypeScript	40	539	305	2608
Python	43	737	695	2240
Markdown	5	331	0	1233
YAML	7	69	33	418
JSX	5	71	36	409
SVG	3	1	1	76
JavaScript	4	11	13	69
Dockerfile	4	51	65	65
Text	1	0	0	22
HTML	1	1	2	13
DOS Batch	1	0	0	7
INI	1	0	0	4
SUM:	119	1811	1150	12823

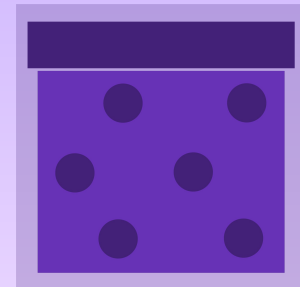
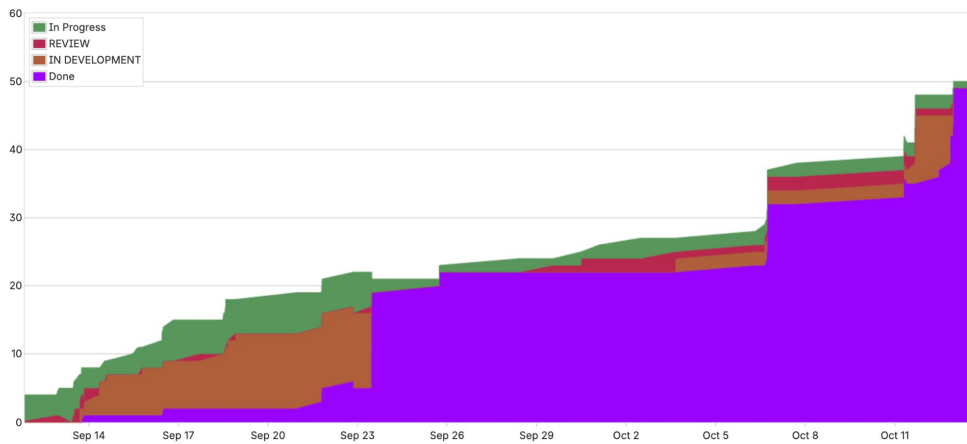
- Average Cyclomatic Complexity = 3.4

Total nloc	Avg.NLOC	AvgCCN	Avg.token	Fun Cnt	Warning cnt	Fun Rt	nloc Rt
1137291	12.0	3.4	70.2	56098	1253	0.02	0.29



Code frequency

Additions and deletions per week



Thank you!

