Team AKA Wolfgang: Andrew P., Kosta D., Aaron G
              ↑   ↑ PM,   ↑ Handling
         Handling   handling   front-end
         Sql   Python

Pages:
- Login
  ↳ Home
  ↳ Explore
  ↳ Create

Expected Shipment Date:
2-3 weeks, ~Nov. 20

Post-MVP features
- Liking/saving/sending blogs & messaging
- Anonymous blogs (share & hide ver)
- Comments on blogs
- Recommended blogs based on tags
- Follow users
  → Have a separate feed for "Following" posts

Login page

- Credentials
  - Sql3 used to store username & password
  - Use GET request to maintain privacy

- Sql table(s)

              *unique

id	username	Password
22585	apialetsky	Pass123
⋮	⋮	⋮

Enter Username: [   ]
Enter Password: [   ]

Home Page

- Display previews of all of your blogs.
  → Fetches every row of Sql table of your blogs
    & displays meta info.

- Similarly, it will display blogs you've liked

- Nav bar - Home, Explore, Create, Logout

- Sql

T1
id	posted blogs	liked blogs	Drafts
22585	['applesauce',...]	['hamsters', 'humans',...]	['straws', ...]
⋮	⋮	⋮	⋮

T2
id	posted blog title	date posted	Content
22585	"Straws"	null → draft	{"title": "Straws!", "imgs": [straws.png],...}
⋮	⋮	⋮	⋮

& same for liked posts

Home | Explore | Create | | Logout
Welcome ___usr___ !
Your blogs   click to visit
Your Likes

Home | Explore | Create | | Logout
Title by Usr
Title by Usr

Home | Explore | Create | | Logout
template 1 | template 2 | template 3
TITLE
Subtitle
text
Img
Add text section ⊕

Notes from Gallery Walk
> Editing blogs
  ↳ separate database for each blog w/ history, changes, versions
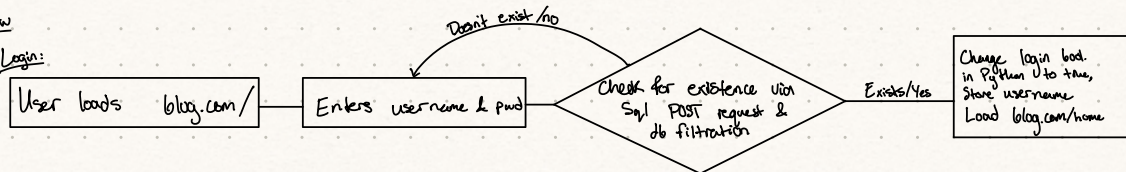> Flow Charts
> Backend
> Flow Charts

Notes from Feedback
> Specificity in SQL tables
> Component map
> POST, not GET, for privacy
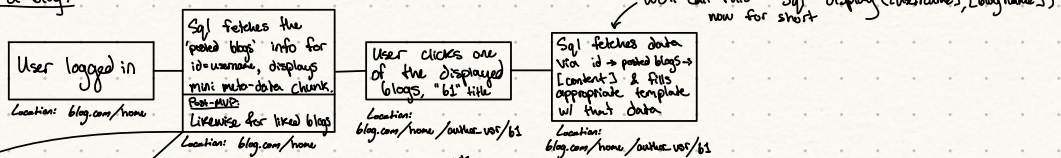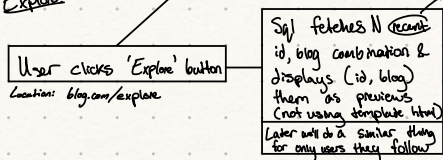> Unique primary values in Sql

---

Revision Additions:

Flow

Login:

```
┌──────────────────┐      ┌──────────────────┐           Doesn't exist/no
│ User loads        │─────▶│ Enters username  │◀─────────────╮
│ blog.com/         │      │ & pwd            │              │
└──────────────────┘      └──────────────────┘              │
                                    │                        │
                                    ▼                        │
                          ╱◇────────────────◇╲               │
                         │  Check for existence │            │
                         │  via Sql POST request│────────────┘
                         │  & db filtration     │   Exists/Yes    ┌──────────────────┐
                          ╲◇────────────────◇╱ ──────────────────▶│ Change login bool │
                                                                  │ in Python 0 to true,│
                                                                  │ Store username    │
                                                                  │ Load blog.com/home│
                                                                  └──────────────────┘
```

Logged in:

Open a blog:

```
┌──────────────┐     ┌──────────────────────┐     ┌──────────────────┐     We'll call this "Sql display ([username], [blog name])"
│ User logged  │────▶│ Sql fetches the       │────▶│ User clicks one   │      now for short
│ in           │     │ 'posted blogs' info for│     │ of the displayed  │
└──────────────┘     │ id=username, displays  │     │ blogs, "b1" title │     ┌──────────────────┐
Location: blog.com/home│ mini meta-data chunk.│     └──────────────────┘────▶│ Sql fetches data  │
                     │ Post-MVP:             │     Location:              │ via id → posted blogs→│
                     │ Likewise for liked blogs│   blog.com/home/author_us1/b1│ [content] & fills │
                     └──────────────────────┘                            │ appropriate template│
                      Location: blog.com/home                            │ w/ that data      │
                                                                         └──────────────────┘
                                                                         Location:
                                                                         blog.com/home/author_us1/b1
```

Explore:

```
                            needs work
┌──────────────────┐     ┌──────────────────────┐
│ User clicks       │────▶│ Sql fetches N recent  │
│ 'Explore' button  │     │ id, blog combination &│
└──────────────────┘     │ displays (id, blog)   │
Location: blog.com/explore│ them as previews     │
                         │ (not using template html)│
                         │ Later will do a similar thing│
                         │ for only users they follow│
                         └──────────────────────┘
```

Create:

```
┌──────────────────┐   Location: blog.com/auth_us1/bx   ┌───────────────────┐   ┌──────────────────┐   ╱◇────────◇╲   ┌──────────────────┐
│ User clicks       │──────────────────────────────────▶│ ask for           │──▶│ We will hard code │──▶│user fills│──▶│ Presses post  │──▶│ Set post date     │
│ 'Create' button   │                                   │ title ⇒ bx        │   │ front-end input forms│  │out       │  │ (instead of save)│   │ (now publicly avail)│
└──────────────────┘                                    └───────────────────┘   │ in the arrangement that│ └──────┘   ╲◇────────◇╱   └──────────────────┘
Location: blog.com/create                                        │             │ they'll be displayed)│                │ saves        Location: blog.com/home
                                                                 ▼             │ Display these on the new│              ▼
                                                        ┌───────────────────┐  │ page              │        ┌──────────────────┐
                                                        │ Create new Sql entry│ └───────────────────┘        │ Leave posted date │
                                                        │ T1/id/drafts/bx   │  Location: blog.com/auth_us1/bx│ as null           │
                                                        │ &                 │                               └──────────────────┘
                                                        │ T2/id/bx          │                               Location: blog.com/home/
                                                        └───────────────────┘
```