

THE STAGE MANAGER'S BOOK

Presented By

Hannah Brousseau, Logan Brower, Kaitlyn Bryant,
Ben Fioresi, Liam Tucker

PROJECT FOCUS



Create a one-stop-shop
for everything needed
for a theater production



Create and join hubs



Upload documents
(scripts, photos, etc.)



Create a profile,
complete with résumé
and production roles

SPRINT GOALS

- Familiarize ourselves with our tech stack
- Flesh out the skeleton for our website
- Create dummy accounts and data to populate our database
- Create ‘login’ and ‘create account’ pages



MAJOR SPRINT ACHIEVEMENTS

- React client and Node server set up
- MongoDB database implemented
- Working login page
- Ability to upload files



PROJECT & SPRINT BACKLOGS

A	B	C	D	E	F	G	
1	ID	Story	Time Estimate (hours)	Priority (1-5)	Assignmen	Sprint Num	Status
2	1	Developed frontend page for login	5	1	Logan	1	Completed
3	2	Design MongoDB database to store user credentials	2	1	Liam	1	Completed
4	3	Develop frontend page for profile creation	4	1	Kaitlyn	1	Completed
5	4	Design MongoDB database to store profile data	5	1	Liam	1	Completed
6	5	Set up react client	4	1	Ben	1	Completed
7	6	Set up node server	4	1	Ben	1	Completed
8	7	Integrate create account, login, and forgot password pages to current react app	5	1	Hannah	1	In Progress
9	8	Develop frontend page for account creation	2.5	2	Hannah		In Progress
10	9	Allow users to create and edit profile and save changes to a new page	8	2	Kaitlyn		In Progress
11	10	Encrypt all MongoDB data	3	2			
12	11	Set up google drive uploads	8	2	Ben	1	Scrapped
13	12	Set up google drive folders and reading files	8	2	Ben	1	Scrapped
14	13	User access restrictions + access denied / not logged in page	12	2	Logan		
15	14	Rewire the web app as the login page being the first thing a user sees and nav after login	6	2	Hannah		In Progress
16	15	Develop MongoDB database to store hub information	5	2			
17	16	Develop frontend container for hubs	12	2	Logan		In Progress
18	17	Allow users to create and manage hubs	15	2	Logan		
19	18	Allow users to create Access Codes for Hubs	5	2			
20	19	Develop "Team" page for hub	2				
21	20	Develop RBAC database for users of each hub	10	2			
22	21	Implement GridFS for MongoDB file uploads	8	2	Ben	1	Completed
23	22	Error handling for file uploads	4	2	Ben		In Progress
24	23	Implement Google Sign-In for the login/create account pages	15	2	Hannah		In Progress
25	24	Integrate professional headshot, resume into profile page	3	2			
26	25	Verify login, profile data input to avoid NoSQL Injection Attacks	5	2			
27	26	Protect against Cross-site scripting attack when using express	5	2			
28	27	Convert system for identifying current state from local storage to sessionids.	10	2			
29	28	Design frontend login page	2	3	Hannah	1	Completed
30	29	Develop CSS file of styles used throughout webapp	10	3			
31	30	Design frontend forgot password page	2	3	Hannah	1	Internal Testing
32	31	Connect users' updates of own profiles to backend DB	2	3	Kaitlyn		
33	32	Design frontend create account page	2	3	Hannah		Internal Testing
34	33	Set up google docs integration	16	3	Ben		Backlog
35	34	Develop "Script" Frontend to display static script from a file	20	3			
36	35	GridFS display list of files and allow users to download	8	3	Ben		Backlog
37	36	Modify Frontend files to display loading icon until all profiles load, then	5	3			
38	37	Develop simple helper functions for file uploads/downloads	6	3	Ben		In Progress
39	38	Set up google calendar integration	8	4	Ben		Backlog
40	39	Send notification to user based on google calendar integration	10	4			
41	40	Develop Announcement Board for each hub	20	4			
42	41	Memories page file setup	5	5	Ben		Backlog
43	42	Memories page frontend	8	5			
44	43	Develop Chat feature associated with each user	40	5			
45	44	Replace branding for website (icons, logos, etc.)	6	5			

- Develop pages for login and profile creation
- Design MongoDB database to store user credentials
- Design MongoDB database to store profile data
- Set up react client and node server
- Integrate create account, login, and forgot password pages to current react app
- Implement GridFS for MongoDB file uploads

LESSONS LEARNED

- Decided against using Python for middleware
 - Transitioned to using React for the frontend and Node for the backend
- Had to change gears on the Google Drive API implementation
 - Transitioned to using GridFS with MongoDB
- There were some issues with client-server communication
- Division of tasks could be improved
- Git issues
 - Merge conflicts
 - Unstable commits



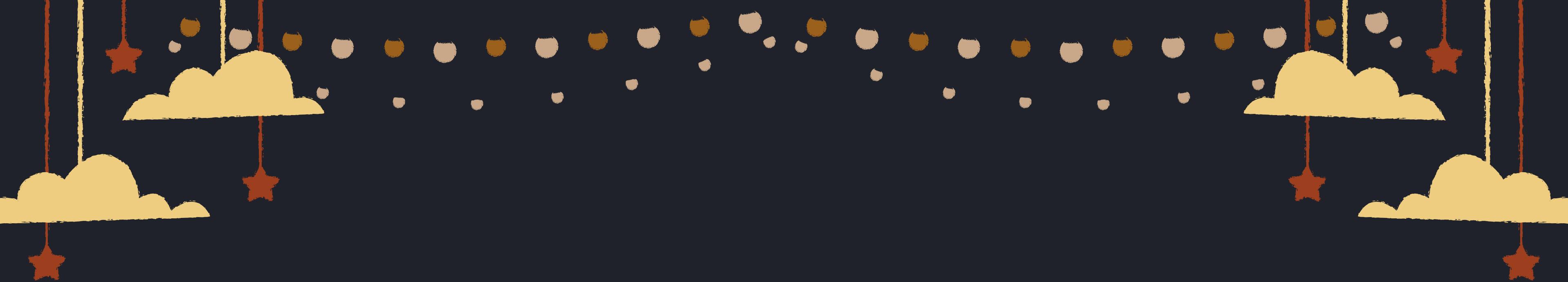
LIVE DEMO



A screenshot of a Microsoft Visual Studio Code interface. The title bar shows "stage-managers-book". The Explorer sidebar on the left lists a project structure: "STAGE-MANAGERS-BOOK" containing "client", "figma-items", "front-end", "server", ".gitignore", and two "README.md" files. The main editor area displays the content of "README.md". The terminal at the bottom shows a command-line session:

```
letuo@LAPTOP-MVLGQ09E ~/OneDrive - The University of Alabama/495CS/v2/stage-managers-book/client
$
```

The status bar at the bottom indicates "Ln 1, Col 1" and "Spaces: 4".



QUESTIONS?

