

- **Title:** Fakebook.com
- **Who:** Ethan Shuler, Jared Marks, Jared Myers, Kayla Kloster, Joseph McSoud
- **Project Tracker:** Jira
 - <https://notrussia.atlassian.net/projects/FAK/issues>

The screenshot shows the Jira interface for the 'Fakebook' project. The left sidebar contains navigation options like 'FAK board', 'Backlog', 'Active sprints', 'Reports', 'Releases', 'Issues and filters', 'Pages', 'Components', 'Add item', and 'Project settings'. The main area displays the 'FAK Sprint 2' board. It has three columns: 'TO DO', 'IN PROGRESS', and 'DONE'. Under 'TO DO', there are two sections: 'Newsfeed Layout (Front End) (Design)' with tasks FAK-12 and FAK-13, and 'Newsfeed Layout (Front End) (Coding)' with tasks FAK-22 and FAK-23. Under 'IN PROGRESS', there are tasks FAK-9, FAK-10, FAK-11, FAK-19, FAK-20, and FAK-21. The 'DONE' column is currently empty. The board also shows a search bar, filters for 'Only My Issues' and 'Recently Updated', and a status bar indicating '9 days remaining' and 'Complete sprint'.

- **Code VCS (has readme.md):** <https://github.com/EthanShuler/Fakebook-Code>
- **Milestones VCS:** <https://github.com/EthanShuler/Fakebook-Meeting-Logs>
- **Meetings VCS:** <https://github.com/EthanShuler/Fakebook-Meeting-Logs>
- Screenshot of each member's contributions throughout the semester from Git repo

Jared Marks
Jared Myers
Kayla Kloster

The screenshot shows the GitHub repository 'Fakebook-Code' on the 'master' branch, specifically the 'views' directory. It lists five files with their descriptions:

File Name	Description
404.ejs	refine 404 error page
fakebook.ejs	name added to post
login.ejs	fix offset logo view
profile.ejs	final touches
register.ejs	final touches

Joseph McSoud

Branch: master ▾ [Fakebook-Code](#) / db /



EthanShuler working on passport and session

..



index.js

working on passport and session

Ethan Shuler



.gitignore

delete nodeModules from git



README.md

create Readme file



app.js

alter app.js to work with heroku



package-lock.json

adding connect postgress for session storage



package.json

adding connect postgress for session storage



testcases.txt

describe how testing was done

Branch: master ▾

[Fakebook-Code](#) / routes /



EthanShuler change register system and add profile pic

..



index.js

change register system and add profile pic



oldIndex.txt

name added to post



users.js

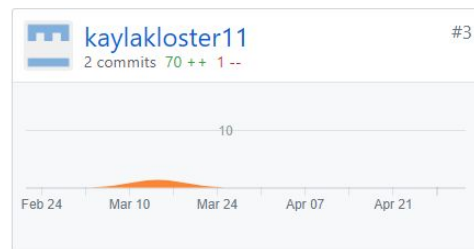
working on passport and session

Commits

Feb 24, 2019 – May 5, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits



- **Deployment:** <https://fakebook-3308.herokuapp.com/login>
- Fakebook-code VCS tagged with “final submission”
- Fakebook-code VCS has README.md
- Fakebook-code repo structure:
 - /db - contains database info for a local db
 - /public - contains css files, js files, and images
 - /routes - contains index.js, where most of the logic for the application lies
 - /views - contains ejs files
 - .gitignore - ensure that node_modules is not pushed to git and heroku
 - app.js - set up the server using nodeJS
 - package.JSON - lists the nodeJS packages and development script
 - testcases.txt - text document describing how we tested code.
- To build: download the project files. Enter the directory in terminal. Use the command “npm install” to install the nodeJS packages. Type “npm run dev” in the command line. <http://localhost:3000/login> will bring you to the login page. Your local database will not be

set up the same as ours, so you would have to setup a session table, users table, and posts table and change several lines of code. We recommend visiting our heroku deployment to actually run the code.

- To run the code, simply visit the production environment on heroku. Because we have set up the code to work on heroku, it would be difficult to run the code locally unless you have the same database as us. You would also have to uncomment and comment several lines of code in `app.js` and `routes/index.js` in order to use a local database.
- To test the code, I have described our testing procedure in the Fakebook-Code repository in a `.txt` file. We used manual testing, so tests were comprised of entering data into fields and ensuring the correct result appeared.