

DEVELOPED BY TEAMCACHE:

Paul Narup

Sebastian Kurpiel

Andrew J Rorher

Daoud Shafique

Table of Contents

Abstract:	2
Project Narrative:.....	2
Pre-Planning Phase:	2
Pre-development/ Planning Phase:.....	5
Development Phase:	6
Testing & Conclusion:	13
Members and Contributions	14

Abstract:

InspireLabs is a place where creators and groups can come together to share, plan, and create projects or events. Our application allows users to easily create and maintain their projects. The motivation to create InspireLab came from our dislike and discomfort of using trello as a project managing solution. Over the course of the semester we strived to create an application that we as developers, project managers, and students would want to use. This report shows in detail the process from pre-planning to final release. Also included are personal narratives of our thought process and what we imagine the final release look and do. Enjoy and prepare to be “inspired”.

Project Narrative:

The idea was to create a project management system that combined the best of Jira with the best of trello. However, as the project brainstorming went on we found out that there was a lack of software or an application that allowed users to easily collaborate on projects. InspireLabs solution is to make the life of group projects life easier by synthesizing all relative material in the same place. An easy way of visualizing this project is by thinking of a folder. One of the main functionalities of a folder is that you can store things in them, be it a file or more folders. Just as such, our solution offers users a similar way to organize ideas and smoothen group projects. Files in this sense would be the information that you want to share with your group etc., folders are the containers or subcategories to make your organization easier. Our take on folders are mosaics and within your mosaic you can have tiles that store your information. We didn't want to replace other services like Google drive, docs, or photos because Google already does a great job with those services. Our intent was to create a platform where users/groups didn't have to have all their information spread around, instead it would be all in one location for easy navigation.

A default example is a group of friends' Summer 2018 plans, within their tile they can include a google maps with the locations they want to visit, a list of movies they want to watch, and a google photos album where they can share their photos. Without our application the users would have to create a separate repository in which they store the links to those locations i.e. a google docs sheet or some sort of online document. Our application allows for the group to have repository which is always available online and that the group can easily change and edit.

Pre-Planning Phase:

Brainstorming:

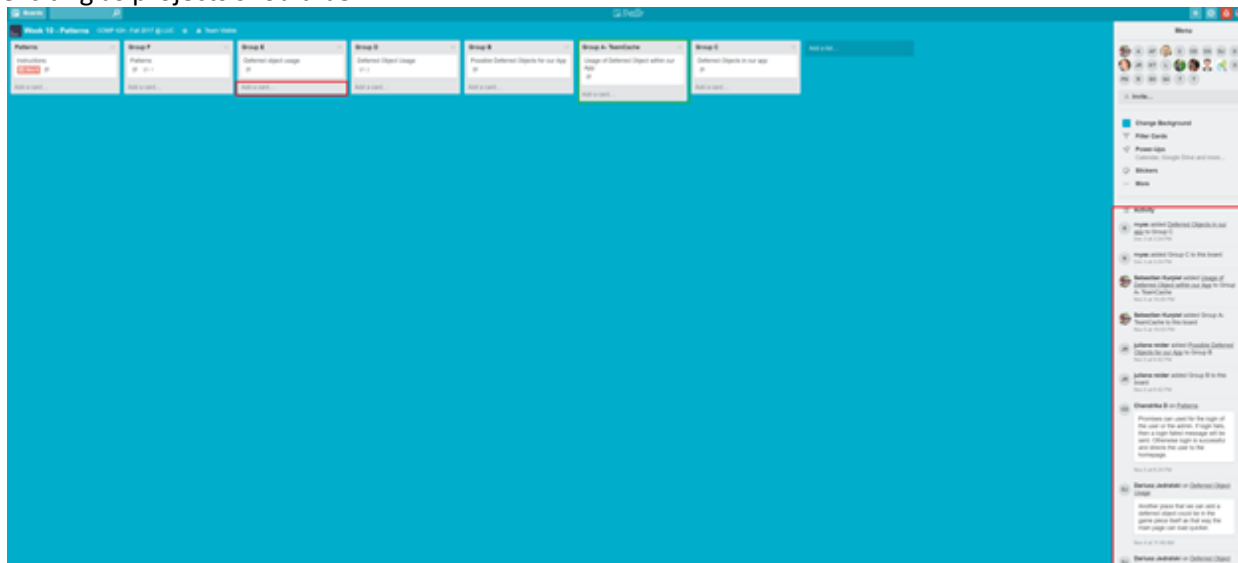
The group discussed various ideas at this point, the main three were: a grocery list app that allowed the user to make list and to scrub items, a banking app that did mock banking tasks, and a Pinterest-like application that allows the user to create list with anything they like. Building up from the last idea we discussed, over slack, what wanted the application to be and how it could benefit the user. We decided that the application should be around projects and events, rather than like Pinterest's lists and ideas. During this discussion we discussed how much we disliked using trello and

that there had to be a better application to plan a project. This is where the project focus came and in and we decide this was the main route we wanted to take.

Case study:

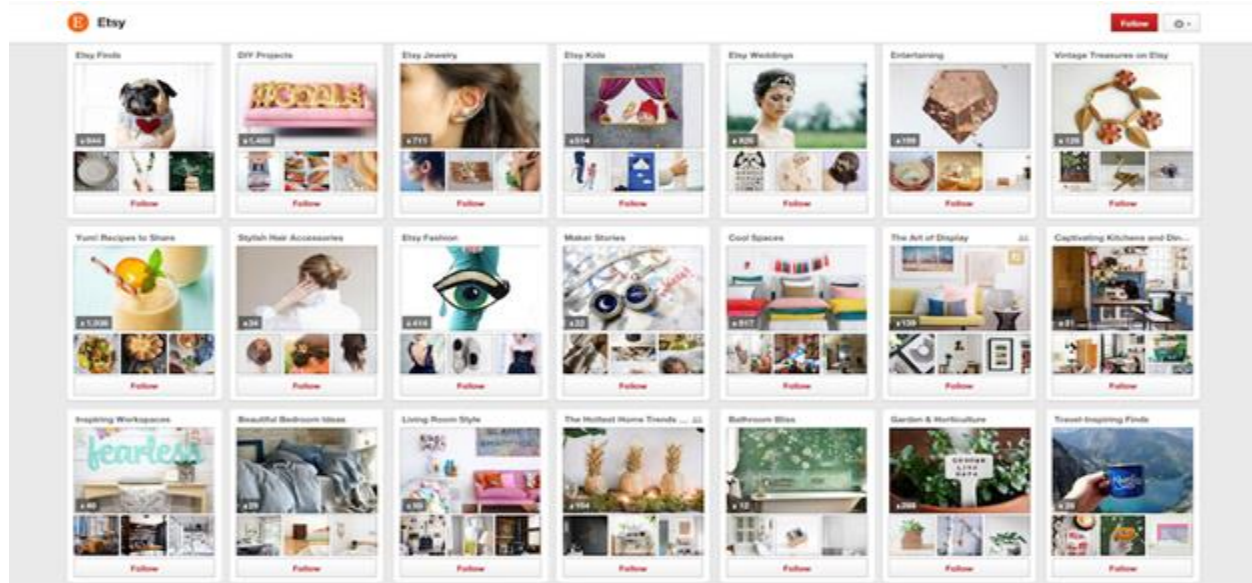
At this point the group created a brief case study of both trello and Pinterest. This is what we found about both applications which we disliked and would ideally prevent in our solution

Trello: Trello is a software for project management and assigning tasks, it can also tie into software like Jira. The thing we liked from trello was the tile system. The system allows for easy to read display of information and allows for more in-depth information if the user decides to click on the tile. Another feature of trello that we enjoyed was the activity feed, which is an easy way for the user to see updates from the lists they subscribe to. We saw that this could be used to update the user about any new additions to projects they follow. The thing we disliked about trello was the linear system, which doesn't always work for all projects and events. Also, the user experience felt clunky and wasn't as fun and exciting as projects should be.



(Highlighted: Green are things we like, also the activity feed. Red is what we disliked)

Pinterest: Pinterest is social media platform that allows users to create lists of products, items, ideas and basically anything they want. The thing we liked about Pinterest were the user experience. It felt easy to use and wasn't as clunky as trello. We also liked how the UI looked, it was less intimidating than trellos's and looked very modern. The things we disliked where the lack of the ability to add certain files. Pinterest now only supports images, therefore making less useful for groups to plan projects and events. Another feature that Pinterest lacked was group functionality, at the moment only one user could update the list and it would then notify the other group members when the list was published.



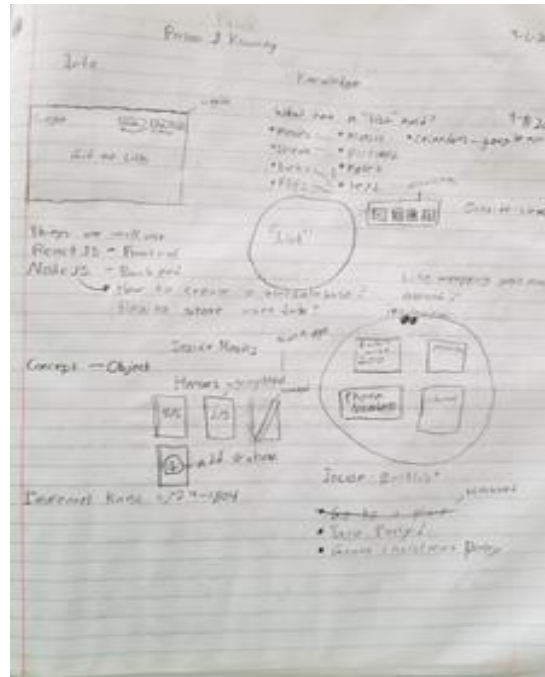
User functions

Discussed in this phase was the user stories that the final application had to encompass.

Functions a User should be able to do:

- a user should be able to register
- a user should be able to login
- a user should be able to see his or her project
- a user should be able to create a project
- a user should be able to edit a project that they have permission to edit
- a user should be able to delete a project that they have permission to delete
- a user should be able to share their project
- a user should be able to look at other projects
- a user should be able to follow/favorite a project

Pre-development/ Planning Phase:



(An early planning of the UI, drawn in Sebastian's Philosophy Notebook)

The above image shows the early planning phase of what would become InspireLab. During that stage the application was geared towards a more non-project approach. Also shown is the first implementation or idea that would evolve to the tile/mosaic folder design. The group's event was encased in a large folder structure, within the structure were subfolders where the user could add more information. In addition, a "scrubbing" feature was introduced that if implemented would allow the user to cross out items, like a to-do list or one of those tv-show tracking apps like trakt.tv. The stack was also discussed, however due to lack of experience and time we decided to scratch the reactJs front-end.

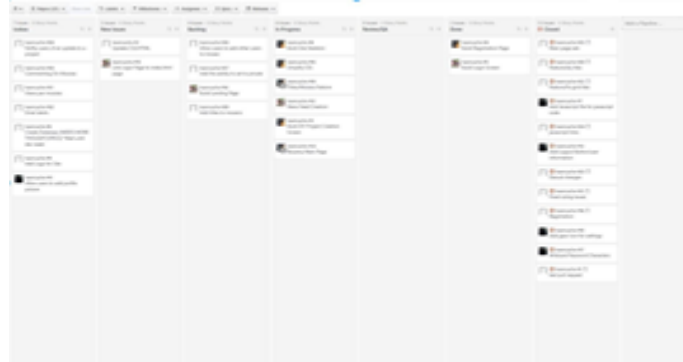
The next step in pre-development was listing out the pages needed to complete the application.

Initial Page List:

- Landing Page: a basic landing page that shows off the product and things it can do.
- Login Page: A page where users can log in to access their "tiles".
- Registration Page: page where users can register for our application.
- Main-Page: the home page of the application, from there the user can view all his or her tiles and created new ones or modify existing ones.
- Title Page: a page where the user can interact with the tiles, take a closer look, add photos, add other files, etc.
- Contact Us Page: a page where the user can send a message to the site owner

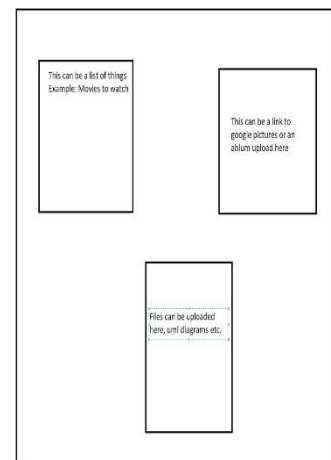
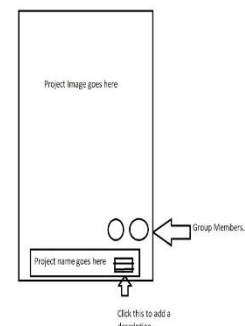
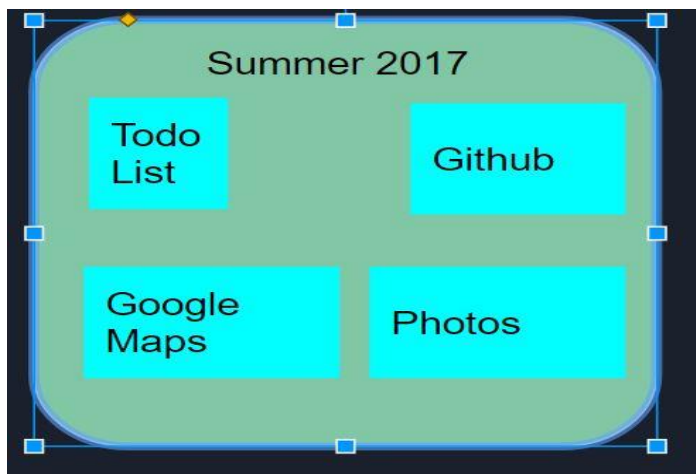
Development Phase:

Armed with our knowledge of CSS, HTML, and JS we struck out to build the application design previously. But before we started coding Andrew and Sebastian assigned the group tasks and goals to complete the week before DEV Week. We used an application called Zenhub up and through dev week to list the task that needed to be done.



(Initial Zenhub board used up to Dev Week)

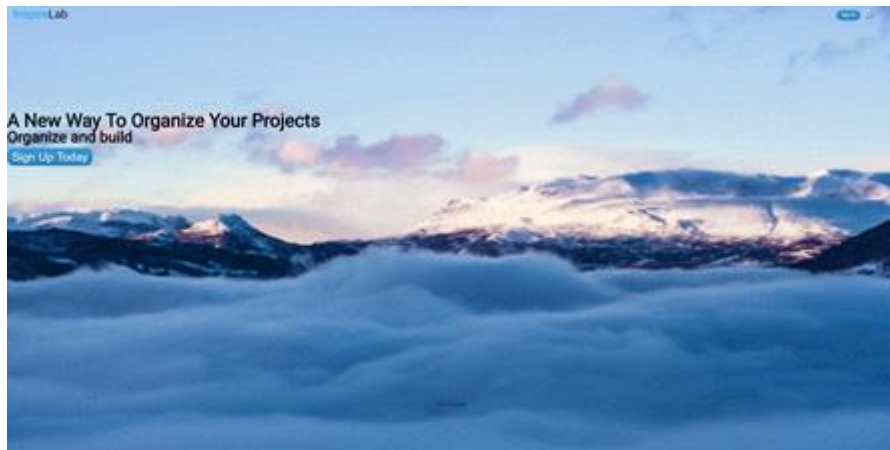
Tile Design:



This is the initial tile design we had from dev week. As you can see it's based off the example described in the project narrative. It is at this point where the idea of mosaics came, tiles within a bigger tile to complete the whole picture of the project.

Pages worked on and Implemented by Dev Week:

Landing- Page:



A simple landing page that the user is redirected to when loading the application. It has links to the login and signup page. Sebastian based it off the current landing page for Netflix. The reason we went with a mountain/cloud gif background was that it's quite calming and our application stores your info in a "cloud" like atmosphere. We planned on improving the page with a list of features that user can do within our app.

Login Page:



A simple login page that logs in the user with an email and password. At dev-week it was a dummy static page. We chose the gif in the background well...because we enjoyed it at the time and we thought it added an element of fun to the project. It's probably the only time the color scheme is broken throughout the project.

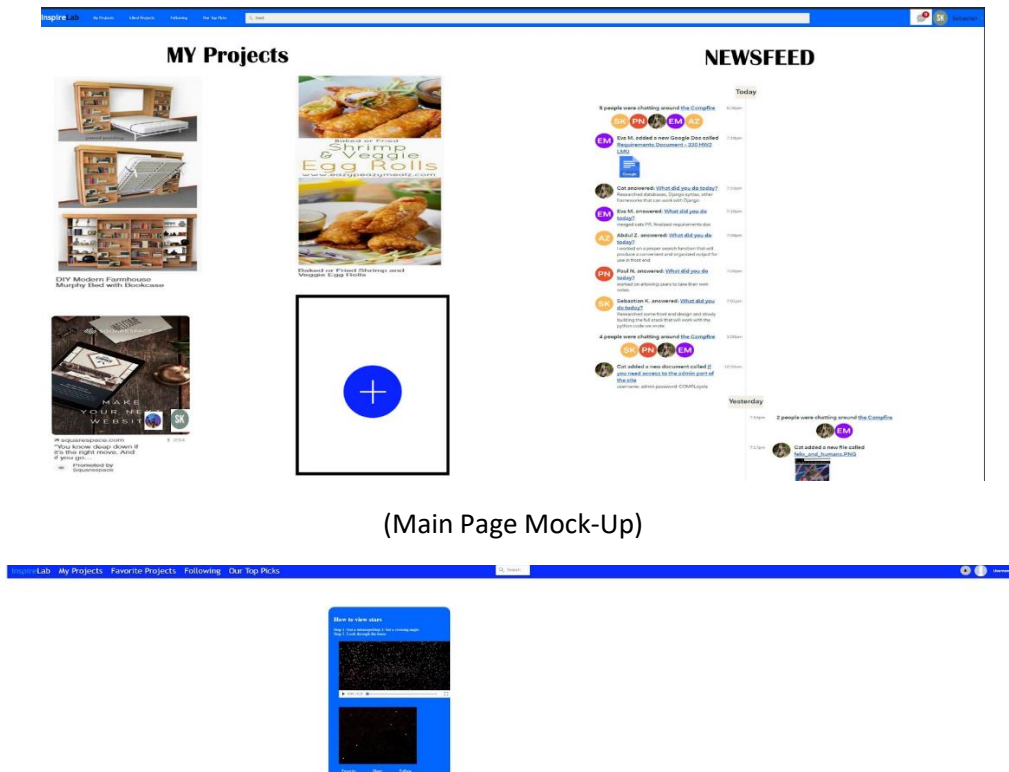
Registration Page:

A simple registration page created so that the user could register for the application. It had a dummy form and was linked to login at the time.



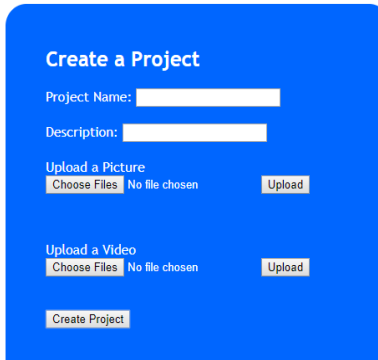
The registration form is titled "InspireLab Register". It contains the following fields: "First Name" (placeholder: First Name), "Last Name" (placeholder: Last Name), "Email" (placeholder: Email), "Password" (placeholder: Password), and "Confirm Password" (placeholder: Confirm Password). A blue "Submit" button is located at the bottom right of the form.

Main Page:



(Main Page Mock-Up)

(Main-Page at Dev Week)



Create a Project

Project Name:

Description:

Upload a Picture
 No file chosen

Upload a Video
 No file chosen

(Create a project tile)

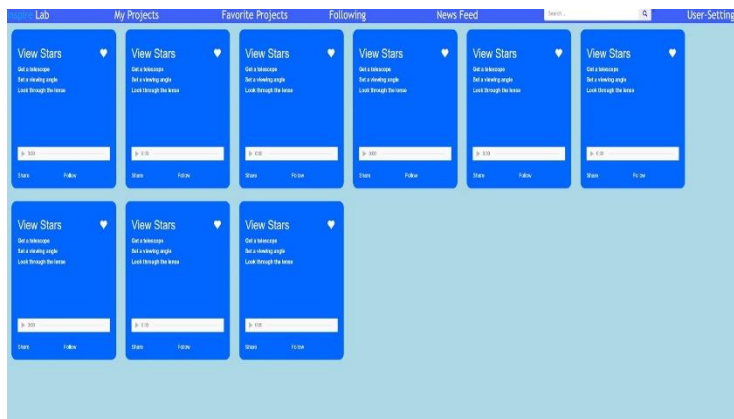
This was the main page of the application at dev week. As you can see we took features such as the top nav-bar, search bar, username, and the tile feature from the mockup. We also implement a dummy tile in which the user could create the project from main page directly. The main goal of this page was to have all the user created tiles displayed toward him, with separate pages for favorite projects, top projects of the week, and projects that you were a collaborator on.

Development After Dev Week:

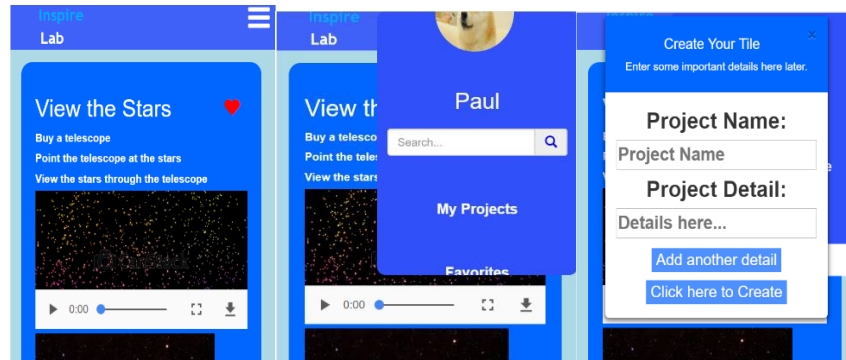
After taking feedback both from Dr. Hayward and the class we continue with the development of the application. We finalize the stack which is NEMP. We are using node.js as the environment, Express as the framework, MongoDB as a mlab instance to store user info, and plain JavaScript, html, and CSS as the front end. Below is what we implemented for the Final Presentation.

Main-Page Redesign:

From the class's comments from Dev Week it was obvious to us that the user interface for the main-page need to be redone. Therefore, we set out to modernize the UI. The new interface is containing lighter colors so that the user is not met with a dark theme that scares them away. To make the top nav-bar less cluttered we went with additional pop-up side nav-bar. The side nav-bar contains the user's settings and the ability to create new projects. We found that this was the best approach because it sets the users focus on the tiles rather than big nav-bar.



Media Queries:



Media queries were used for each page to ensure that mobile users would have a good user experience. The focus of the queries was to make sure everything on the page had the level of visibility necessary for the user to navigate the website effectively.

The goal was to avoid changing the layout of the pages except for the main page. A hamburger menu was added in place of the nav bar buttons. This menu opens a side nav bar which gives the user access to all pages and settings including search, creating tiles, and changing profile pictures.

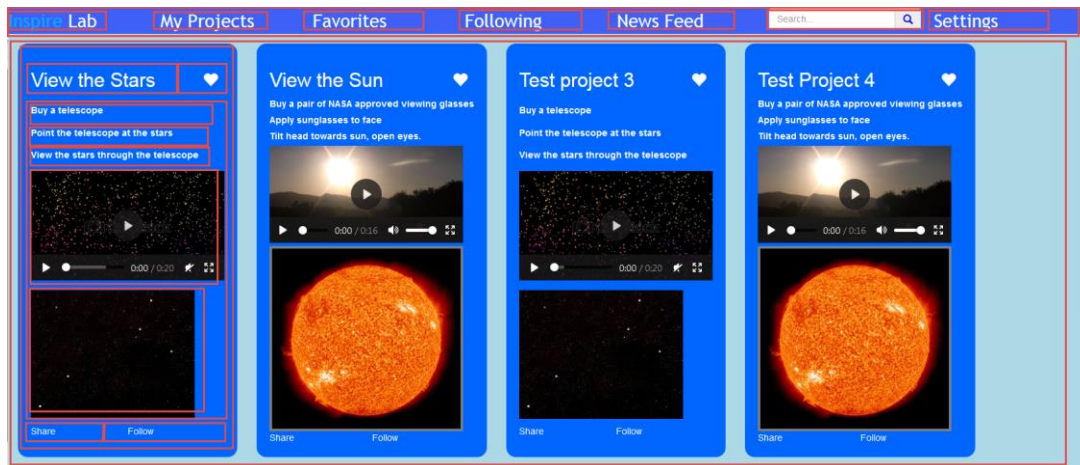
Login:

The goal was to have the users store their login info as well as verify their info when logging in. This was done by creating a passport JS app which was linked to a mongo dB database. The way it worked was that a user would fill in their info in a form upon creating a new account and it would be stored in the designated database. Jcrypt was used to hash the passwords to avoid the risk of getting the passwords leaked. The form took in a valid email address, meaning it had to have the @ symbol along with something.com. When the user would login again, the app would verify the credentials through the database and allow the user to enter if the keys matched, otherwise it would return an error prompting the user to try to fix the error.

Tile display and creation:

Tile:

Tiles are the main focus of InspireLabs. They show user built DIY projects that they can create. The idea is that others can see their ideas to be “inspired” to do a project. The tiles we currently have are proof of concept that users can add media and other details about their projects. The development of these tiles has evolved over the course of the semester. First, the tiles were just raw html and CSS to prove that we could make a “tile” that is the right shape and that can display the project information of the user. The data displayed by these early tiles was hardcoded into the html. Since then, flexbox has been added to the display of each tile and the rest of the site.



Flexbox is a helpful tool when creating a layout. As seen above the top nav is created by flexbox. The nice thing about this is when the screen resizes flexbox automatically adjusts where the content is on the screen. This is helpful because only one version of code is required for both desktop and mobile devices, making maintainability easier. After implementing flexbox to our liking, we never had to make another change regarding the layout of our tiles, flexbox got all the sizing and content wrapping right without the need for intervention.

The next feature we added to our tiles was the ability of users to add new projects and view existing projects. This required that we have a database. We used mongo dB and mongoose combined with ejs which is a client-side templating language to display the project data. This is what led the next evolution of the design. Each project was now created from just one template, further reducing the maintainability effort. Once the EJS template was complete all that was needed is the data created and entered by the user, populated in the database. Through the development process our design from database to the UI was made simpler and streamlined. Here is an example of our template for the projects:

```

<% for(i=0; i<returnobject.projects.length; i++) { %>
  <!-- Each template is a container itself-->
  <section class="il-container-projectCard standardColor">
    <!-- The project information is in articles-->
    <article class="il-item-projectcard">
      <!-- This article has its own container inside but is still seen as one article to the parent container-->
      <section class="il-item-cardtitle">
        <article class="il-title">
          <h2><%= returnobject.projects[i].title %></h2>
        </article>

        <article class="il-item-heart">
          <span class="glyphicon glyphicon-heart" onclick="toggleFavorite(this)"></span>
        </article>
      </section>
      <!-- This article has its own container inside but is still seen as one article to the parent container-->
    </article>
    <% for (j=0; j < returnobject.projects[i].steps.length; j++) { %>
      <article class="il-item-projectcard ">
        <label> <%= returnobject.projects[i].steps[j] %> </label>
      </article>
    <% } %>

    <article class="il-item-projectcard ">
      <video class="cardMedia" controls >
        <source src='<%= returnobject.projects[i].media.video %>' type="video/mp4">
      </video>
    </article>

    <article class="il-item-projectcard">
      <img class="cardMedia" src='<%= returnobject.projects[i].media.picture %>' >
    </article>

    <article class="il-item-projectcard">
      <section class="il-container">
        <article class="il-item">
          <span class="cardLinks">Share</span>
        </article>

        <article class="il-item">
          <span class="cardLinks">Follow</span>
        </article>
      </section>
    </article>
    <!-- The project information is in articles-->
  </section>
<% } %>

```

Outer for loop for each project.

Display the project title.

Loop through each step in each project.

Add location of video and picture associated with the project.

Add helpful links to the bottom of each project

The data behind each of these tiles is stored in a mongo dB schema and model:

```

var mongoose = require('mongoose');

var projectSchema = mongoose.Schema({
  title: String,
  steps: [{type: String}],
  media: {
    video: {type: String},
    picture: {type: String}
  }
},
{
  collection: 'projects'
});

module.exports = mongoose.model('projects', projectSchema);

```

Heroku:

The current version of application is running on a Heroku test server, <https://inspirelabdev.herokuapp.com>. The git the Heroku instance pulls from is separate from the development git so that development and release do not mix. We found this useful towards the end of development where we keep changing things causing other things to bug out or break completely. The team found this a useful and interesting way to test the application for bugs. It also allows us to onboard new users with them having to clone and run the repo locally.

Testing & Conclusion:

Testing the application occur via each developer setting up an account and testing it both locally and with the Heroku release version. If any bugs were found we would make note and fix them. Also, we made sure that the application could run on both android and IOS devices. We found that Paul's media queries worked perfectly and displayed all information correctly

Some things we would have liked to have implement was an activity feed. We found an api, stream. Api, to implement this feature, but due to pushbacks and delays we couldn't find time to implement it fully. A better sharing experience was something that we strive to create, but we spent too much on displaying the tiles that we didn't work in this feature.

In conclusion, our team has team has hit some pitfalls and dead ends however we are proud to say that we have developed an application that we believe, with more development time, could be a solution to our issues with trello. Currently the application has the function to display and store projects which is the core feature needed for a project management system. This encased in more modern and easy use UI has brought us closer to achieving our goal of a better project management solution. The team has also discussed putting in a little effort over the next coming months to truly see if we can create something that could and can compete against other project management solutions.

Members and Contributions

-Sebastian Jan Kurpiel

Role: Co-Project Manager and Full Stack Development

Contributions:

- Planned weekly sprints
- Assigned roles and assignments for team members
- Created Dev week and Final presentation
- Pushed the application onto to Heroku
- Combined the application parts (login, main page, etc.)
- Debugged issues and tested the application

-Paul Narup

Role: Front End Development

Contributions:

- Design of registration page
- Media queries for mobile redesign
- User uploaded profile pictures

-Andrew J Rohrer

Role: Full Stack Developer and Assisted Project Management

Contributions:

- Planned/scheduled what work was required to meet deadlines
- Contributed to dev week and final presentation
- Added ability to create and view projects, made project view compatible with mobile devices (flexbox). Created general styles for font/colors
- Debugged and tested code
- Created project database schema

-Daoud Shafique

Role: Back End Support

Contributions:

- Created user authorization application using Passport js
- Helped integrate login info with mongo dB
- Assisted with PowerPoint and Final Paper