Form Buster
Team Members:
- Daniel Acosta (dacosta2022@my.fit.edu)
- Christopher Demuro (cdemuro2022@my.fit.edu)
- Alex Merino (amerino2022@my.fit.edu)
- Luka Miodrag Starcevic (lstarcevic2022@my.fit.edu)

Faculty Advisor: Phillip Bernhard (pbernhar@fit.edu)

Client: Phillip Bernhard (pbernhar@fit.edu)

Progress of Current Milestone:

Task	Completion	Daniel	Chris	Alex	Luka	To-Do
Implement, Demo Form Tracker	70%	0%	0%	100%	0%	Tie Users to Forms using account logic
Implement, Test Active Forms	50%	0%	0%	100%	0%	Attaching users to new forms
Implement, Test, Undergraduate for Grad Class Form	100%	100%	0%	0%	0%	None
Implement, Test, Demo Form Editor	50%	20%	5%	5%	70%	Finalize stylistic choices and implementation of them
Setup AWS hosting	100%	33%	17%	33%	17%	None
Implement, Demo, Test Form Template Saving	100%	0%	50%	0%	50%	None
Implement, Demo, Test, Signatures	100%	0%	0%	100%	0%	None

Discussion of Accomplished Tasks:

- **Implement, Demo Form Tracker:** During this milestone, the form tracker began implementation, specifically with the frontend. The first pass of how the tracker should look was implemented with dummy data which was hardcoded in. The data used replicated the data which would be normally passed through the backend to be displayed in the frontend. After a visualization was achieved and tested in the frontend and backend, the next goal was to tie the active forms to the users. This included a slight

modification of the schema for the User object as well as the introduction of the new Signature object, which will be described later.

- **Implement, Test Active Forms:** For the Form Tracker to be fully operational, the database for the active forms must be fully functional and there must be code which allows for creation, filling out a new signature, and relocation of the form from active database to completed database when all parties have signed the form and it is completed. So far, the relocation and signature filling parts of the task are complete, while the active form creation is still needing to be finished. The complexity behind the form creation is much higher than previously believed. Immediately attaching the correct parties to the form is the part which will take the longest to complete since it requires the same type of user but for different majors. Such as the Dean of the one school is needed for a student in that school but a different Dean is needed for a different school. This might require a new user type to be created but, after deliberation as a team, the plan will be decided by the next milestone.
- **Implement, Test, Undergraduate for Grad Class Form:** The second form that we have implemented is the *Permission for Undergraduate Student to take Graduate-Level Course* form, which has been utilized by a few of the members in the past few weeks. The form is a useful one to implement early due to its few amount of inputs and the parties involved are very clear. There is also logic to how many people the form is sent to, based on the student's GPA which is a feature that has been implemented. The form has been created and inputted into the database alongside the FERPA form.
- **Implement, Test, Demo Form Editor:** The form editor began its development life cycle with a simple first implementation using the React package Quill. This package allows for embedding a text editor as an html object to allow for users to modify form objects. This was the largest single contribution to this milestone's progress, which allow
- **Setup AWS hosting:** An account was created and a free EC2 server was set up on Amazon's website. With this server we will be able to begin testing the web app on a remote server instead of locally. This would allow for the team to test the active forms in real time, at the same time as separate users. The goal of this task was to just create the server this milestone and then begin testing on the server beginning the next milestone.
- **Implement, Demo, Test Form Template Saving:** The Form Editor was mainly including the frontend implementation. In the backend, the main task was to ensure the updated form is saved to the database. This included making sure that each form that was submitted to the database had different names. The implementation of the saving was in tandem with the form editor and is versatile enough to not need to be changed if the form editor changes.
- **Implement, Demo, Test, Signatures:** Signatures are a new data type created to help with form tracking. The use of signatures will be to help discern who signed a form and when. They hold: the account that wrote the signature, the date, and whether or not the signature has actually been signed. This is also the most sensible way to attach users to a specific form.

Discussion of Contribution of each Team Member:

- **Daniel Acosta:** Tasks contributed to this milestone include implementation of the Undergraduate Permission to Take Graduate Course form. This included translating the current pdf of the form to our system and saving it into the database. Document correctly displays when selected in the form submission window. Assisted in the design/logic of the form editor, helping bug fix minor issues. Partook in the setup of the AWS server. Did minor bug fixing in general across the frontend.
- **Christopher Demuro:** During this milestone, the implementation of the form editor and form template systems were tested in all aspects, including saving and recalling form templates to and from the database as editable templates, and checking the rendering of those templates once they became active templates. Further work was done on testing and debugging the retrieval of active forms, with specific attention given to the rendering and functionality of input fields on those active forms once they had been saved in the form template editor.
- **Alex Merino:** Throughout the milestone, the bulk of the work completed was associated with the form tracker. This included creating the frontend formatting of the form tracker as well as the initialization of the backend to frontend connection of the active forms. Another task was the creation and implementation of the Signature object which was a secondary task created by the ambiguity of the form tracker task. Also, as part of the team, he partook in the setup of the AWS server.
- **Luka Miodrag Starcevic:** Tasks contributed to this milestone primarily included the implementation of the form template editor using Quill, and the database logic required to save and recall editable form templates. Further work was done to ensure that custom field placeholders were able to be saved, as the Quill editor, in its current configuration, does not natively support the addition of input elements.

Milestone 4 Task Matrix:

Task	Daniel	Chris	Alex	Luka
Implement, Test Form Tracking	0%	0%	100%	0%
Implement, Test Active Forms	0%	50%	50%	0%
Implement, Test, Demo Form Editor	50%	0%	0%	50%
Implement, Demo, Test Updating a Form	0%	100%	0%	0%
Implement, Demo, Test Forms in the Inbox	100%	0%	0%	0%
Implement, Demo, Test Adding comments to Forms	0%	0%	0%	100%

Discussion of Planned Tasks:
- **Implement, Test Form Tracking:** Continuing the completion of fully-functional form tracking feature. Users should be able to see active forms in their home dashboard, as well as the progress of each form along its approval process. Tests will be performed to ensure full functionality and stability of the form tracking front-end interface.
- **Implement, Test Active Forms:** Continue developing the ability to submit forms to the database, and brainstorm a solution on how to handle the newly found complexities with attaching required individuals to the forms upon creation. Ensure submitted forms appear in the database with the correct information (submitter, form data, recipients/approvers). Tests will be performed to ensure that all submitted forms are saved to the active form database and can be reliably retrieved.
- **Implement, Test, Demo Form Editor:** Continuing the completion of fully-functional form editor, including support for creating and managing form templates, adding content, including, but not limited to, text, formatting and other content, and saving and retrieving form templates for modification and use. Add form preview to editor so that changes to forms can be seen in real time as they are being made. Tests will be performed to ensure that all form templates are saved and recalled reliably.
- **Implement, Demo, Test Updating a Form:** Implement the updating of an already existing template. Currently, existing templates can be opened in the form editor, but cannot be modified correctly. Further testing and features are required to ensure template modification is working properly.

- **Implement, Demo, Test Forms in the Inbox:** Implement the dashboard retrieval of forms sent to the inbox after completion. This includes form approvals/disapprovals for students/faculty, as well as the notification of new forms that need to be signed in faculty inboxes. Form approvals/disapprovals in the inbox should have an optional attached note explaining the form's outcome. Tests will be performed to ensure that the correct users only see the forms meant for their user level (i.e. no students should receive submitted forms, only faculty).
- **Implement, Demo, Test Adding comments to Forms:** Implement the ability for faculty to comment on submitted forms, explaining why the form was approved/disapproved. Add an option to send notifications of form comments for approved forms to the student who submitted it. All form disapproval notes should immediately send notifications to the student who submitted it. Tests will be performed to ensure that notifications are properly sent to the inbox and that the notes are visible when inspecting the notification.

Dates of meeting:
- 4/18/2025

Client feedback on current Milestone:
See Faculty Advisor Feedback below.

Faculty Advisor Feedback:

- Website frontend:
- Database/ hosting server:
- Develop Data Structures:
- Requirement Document:
- Design Document:
- Test Plan:

Signature: _____ Date: _____

Evaluation by Faculty Advisor:

- Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Daniel Acosta	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Christopher Demuro	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Alex Merino	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Luka Miodrag Starcevic	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

- Faculty Advisor Signature: _____ Date: _____