Form Buster - Milestone 1

Daniel Acosta, Christopher Demuro, Alex Merino, Luka Miodrag Starčević

Milestone 1 Overview

Task	Completion	Daniel	Chris	Alex	Luka	To do
Website frontend	20%	0%	0%	100%	0%	Inbox, Dashboard, Registration, Sign In
Database/ hosting server	100%	0%	50%	50%	0%	None
Develop Data Structures	25%	0%	50%	0%	50%	Implementation
Requirement Document	100%	5%	15%	75%	5%	None
Design Document	100%	25%	0%	50%	25%	None
Test Plan	100%	5%	95%	0%	0%	None

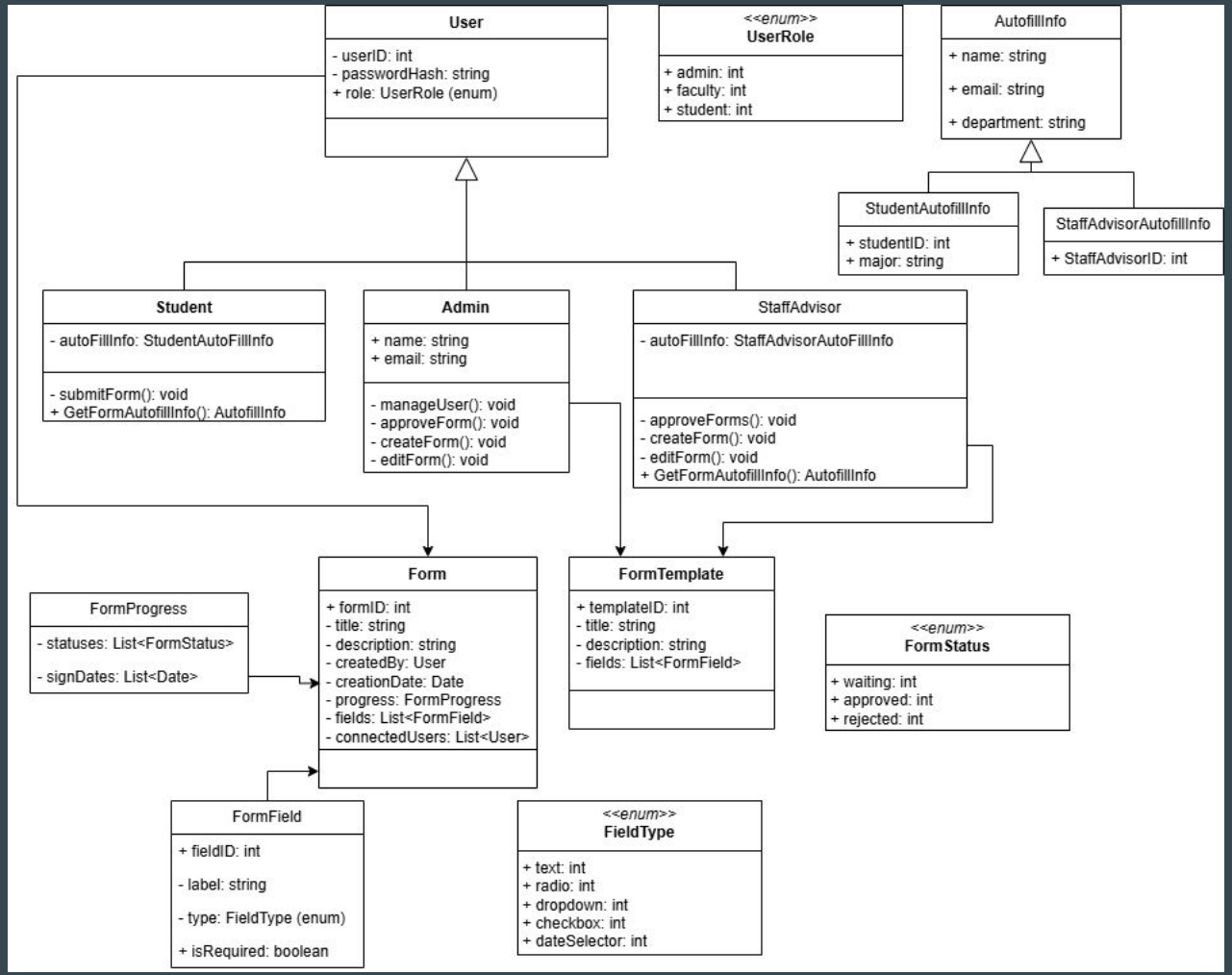
Technical Tools / Resolutions

- After deliberation as a team the best choice for hosting will be EC2 through AWS
  - Cheap
  - Some prior experience
  - Straightforward to use
- Terminology changes
  - Faculty will now be called Staff Advisors

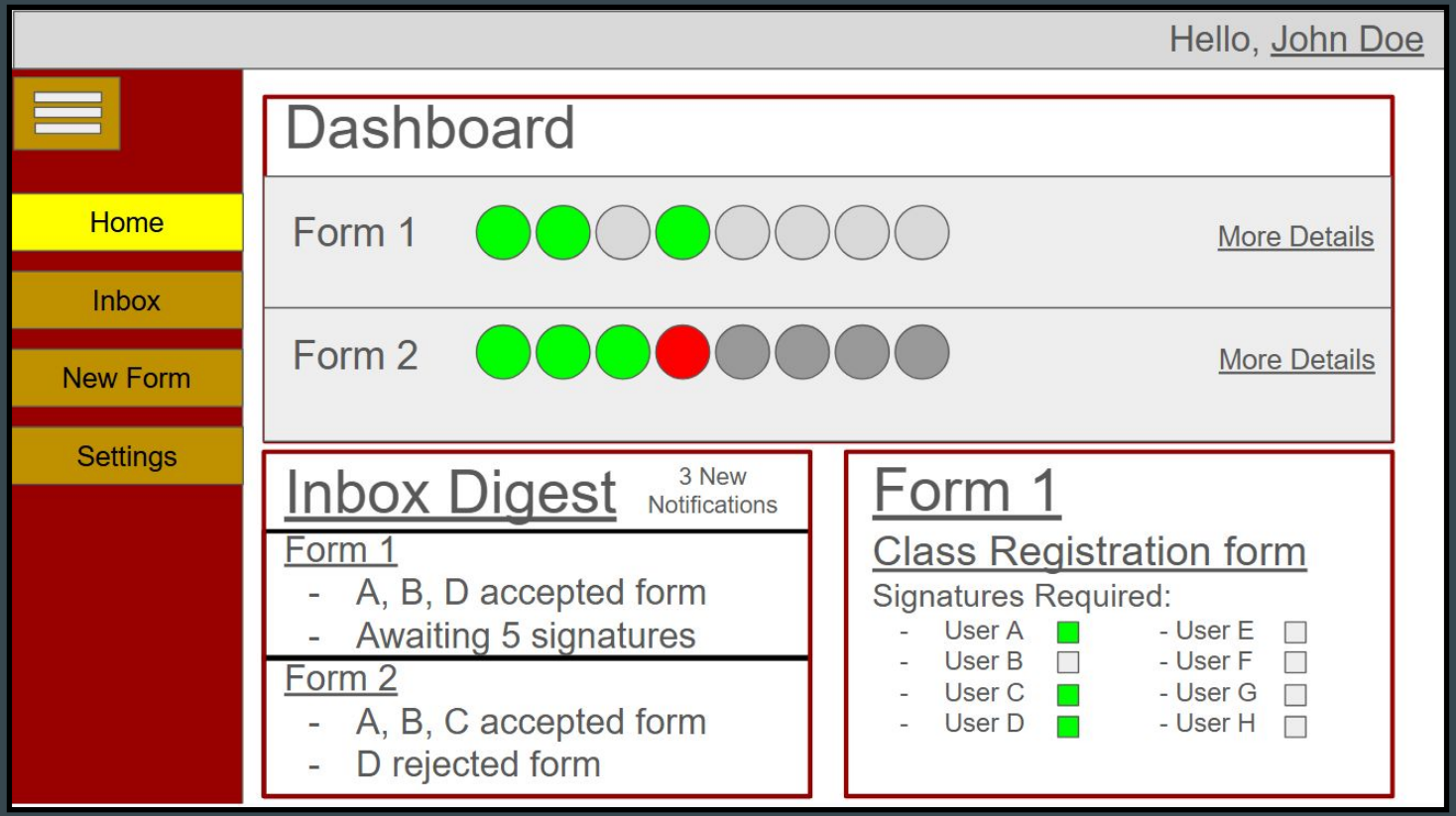
Requirements

- **Form Completion** : The system shall allow all user types to begin and fill out a form
- **Form Tracking** : The system shall allow all users to track a form that requires their input
- **Form Layout Editor** : The system shall allow Administrators to create, modify, (un)publish and delete forms
- **User Registration and Login** : The system shall allow users register/sign in to an account
- **Email Notification System** : The system shall send email notifications to users when a form has been updated.
- **Dashboard** : The system shall display a dashboard with summarized information of their active forms.
- **Inbox** : The system shall list all user notifications in the inbox module

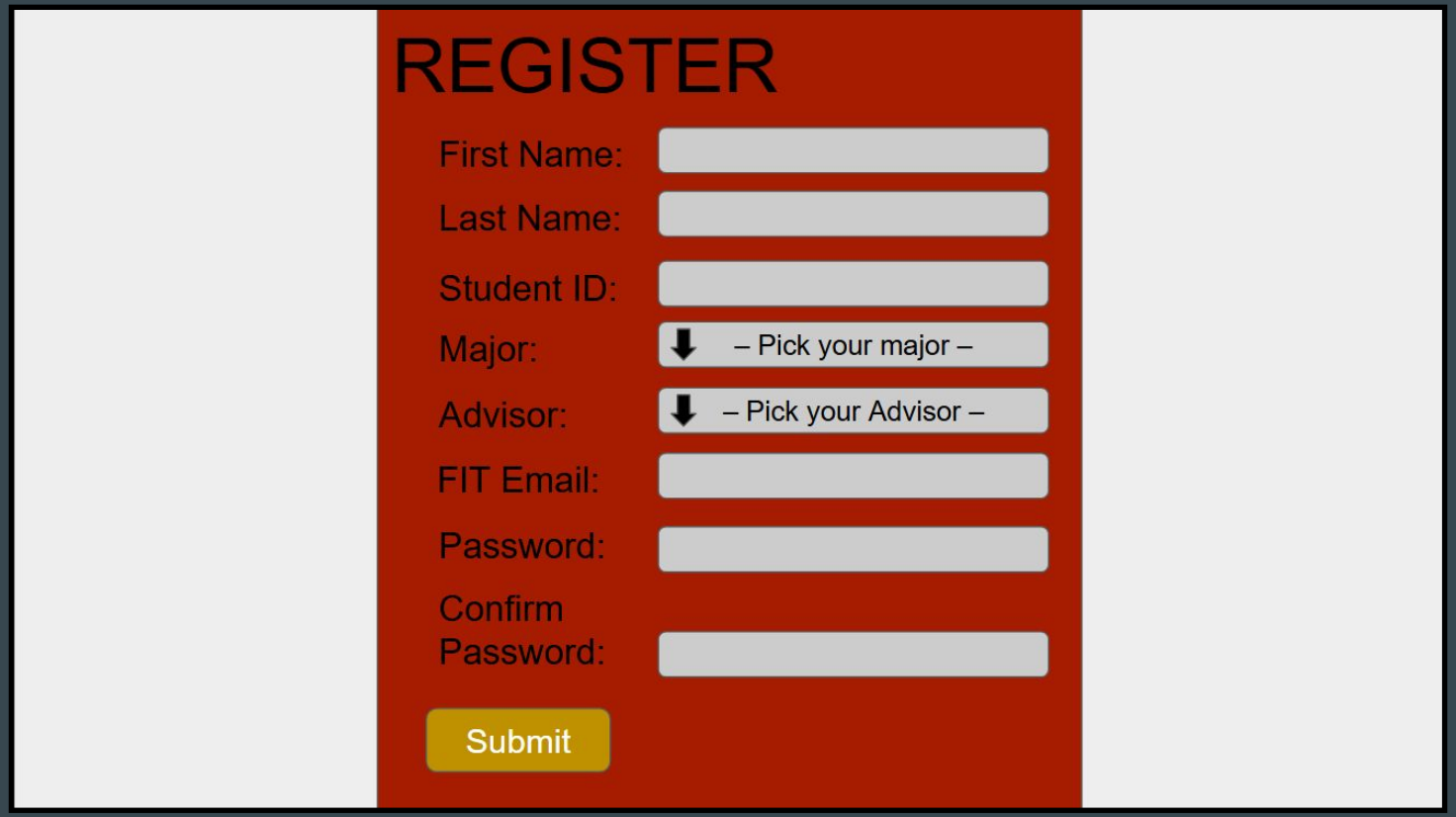
UML Diagram


User
- userID: int
- passwordHash: string
+ role: UserRole (enum)
<<enum>>
UserRole
+ admin: int
+ faculty: int
+ student: int
AutofillInfo
+ name: string
+ email: string
+ department: string
StudentAutofillInfo
+ studentID: int
+ major: string
StaffAdvisorAutofillInfo
+ StaffAdvisorID: int
Student
- autoFillInfo: StudentAutoFillInfo
- submitForm(): void
+ GetFormAutofillInfo(): AutofillInfo
Admin
+ name: string
+ email: string
- manageUser(): void
- approveForm(): void
- createForm(): void
- editForm(): void
StaffAdvisor
- autoFillInfo: StaffAdvisorAutoFillInfo
- approveForms(): void
- createForm(): void
- editForm(): void
+ GetFormAutofillInfo(): AutofillInfo
FormProgress
- statuses: List<FormStatus>
- signDates: List<Date>
Form
+ formID: int
- title: string
- description: string
- createdBy: User
- creationDate: Date
- progress: FormProgress
- fields: List<FormField>
- connectedUsers: List<User>
FormTemplate
+ templateID: int
- title: string
- description: string
- fields: List<FormField>
<<enum>>
FormStatus
+ waiting: int
+ approved: int
+ rejected: int
FormField
+ fieldID: int
- label: string
- type: FieldType (enum)
+ isRequired: boolean
<<enum>>
FieldType
+ text: int
+ radio: int
+ dropdown: int
+ checkbox: int
+ dateSelector: int

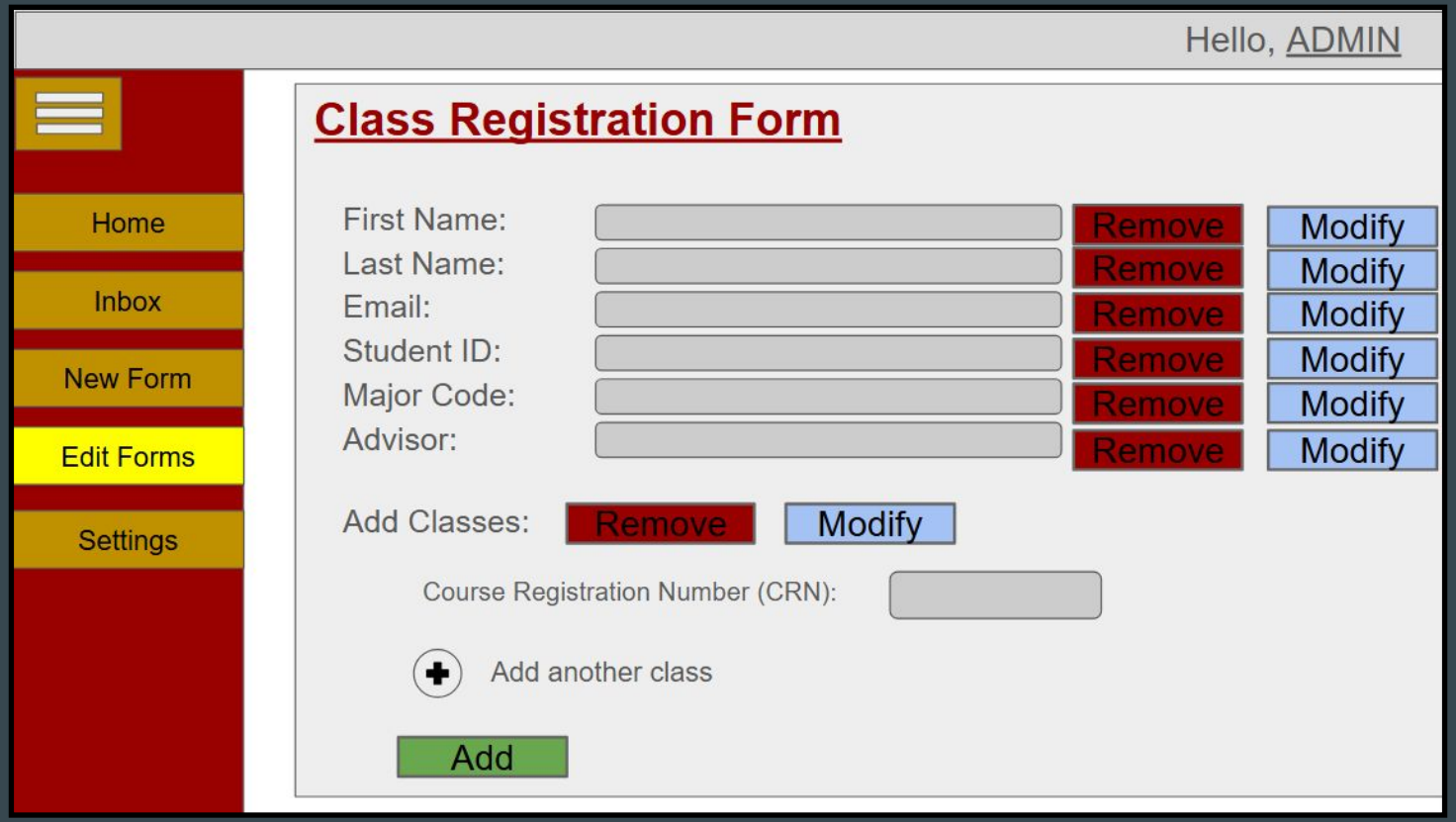
GUI Mockups - Home Page



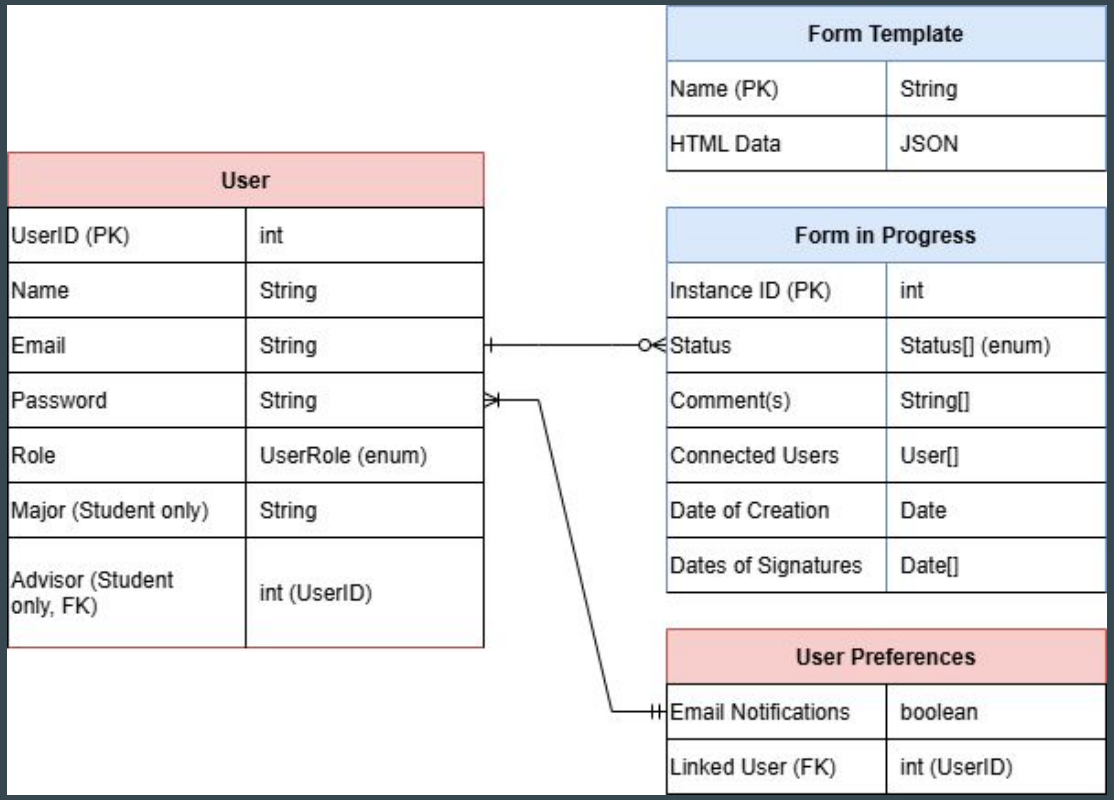
GUI Mockups - Student Registration for App



GUI Mockups - Form Layout Editor



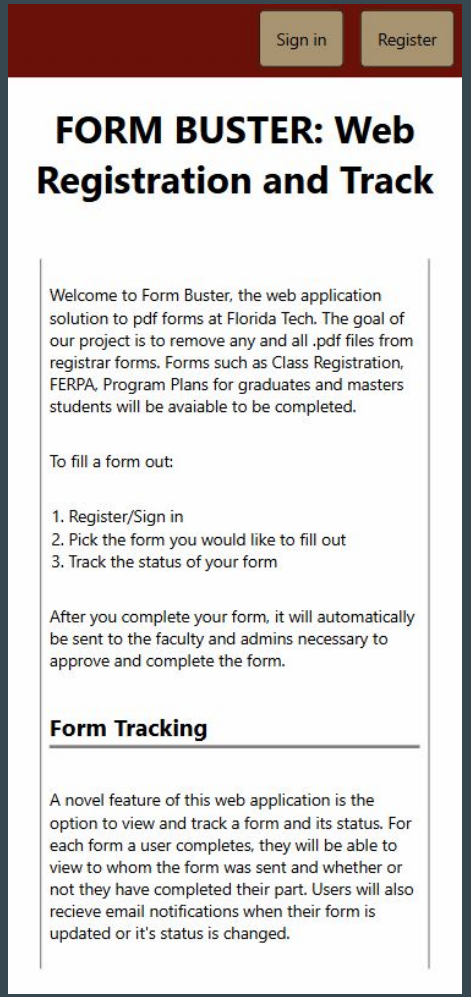
Database Design



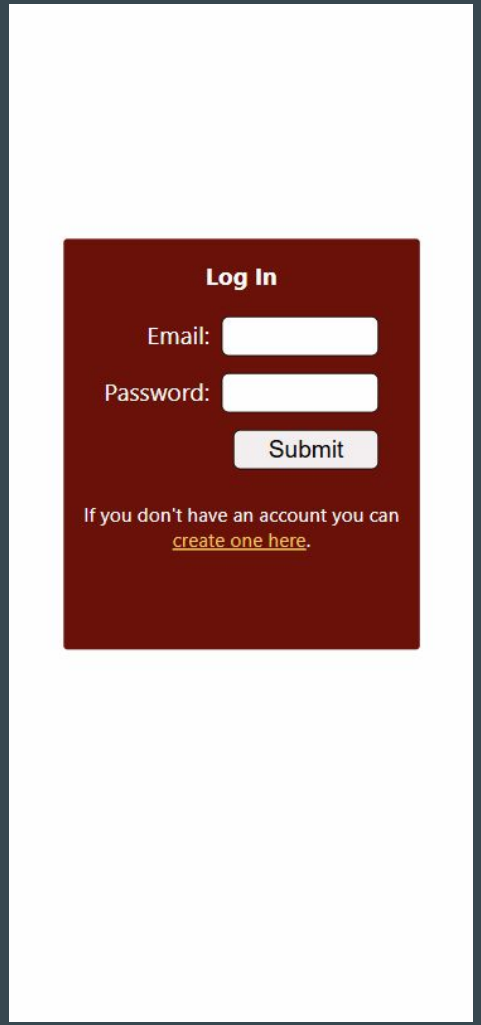
Demo Video

Demo Video

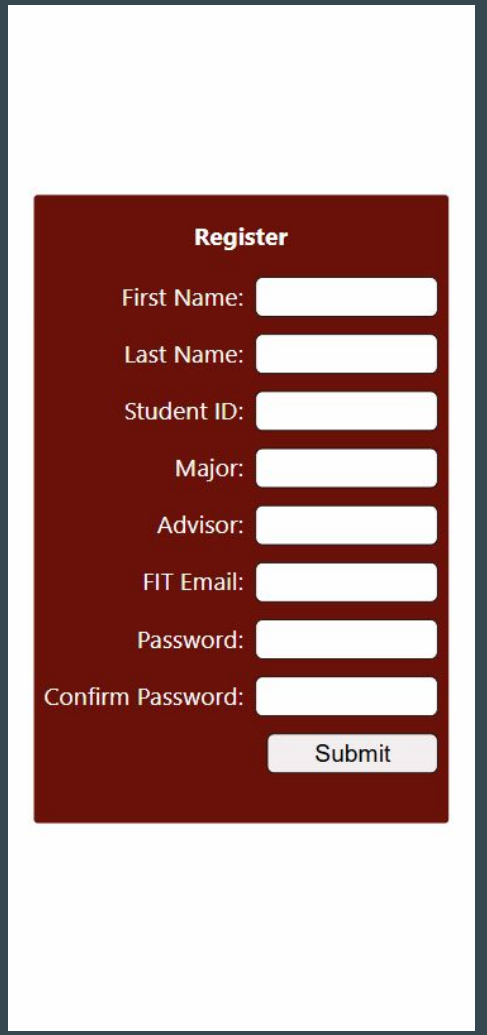
Demo - Landing Page (Mobile View)



Demo - Sign in (Mobile View)



Demo - Register (Mobile View)



Demo - Dashboard (Mobile View)

Testing

- Levels
  - UI/UX
  - Database
  - Interaction
- Test Items
  - User Authentication
  - Profile creation
  - Form Storage
  - Email Notifications
  - Form Template Creation/Modification

Milestone 2

Task	Daniel	Chris	Alex	Luka
Implement Frontend	33%	17%	33%	17%
Implement Routing	0%	100%	0%	0%
Implement Form Editor	25%	25%	25%	25%
Implement User Views	0%	0%	100%	0%
Implement Database	0%	0%	0%	100%
Implement simple form	100%	0%	0%	0%

Any Questions?

...