INTRODUCTION

Development of a web application with a background database that can be used by internal UW College of Engineering (CoE) departments to manage and track requests.

- Department administration lacks tools to manage requests for administrative services, specifically financial transactional duties. Examples include purchasing, reimbursements and travel requests (booking and reimbursements).
- The functionalities of this web application includes:
  - Submission of requests by users with the ability to route for appropriate approvals, upload documents and collection of the information needed to process by fiscal staff.
  - Allow communication between the staff and the requester.
  - Provide status updates on where the request is in the overall process and any communication with fiscal staff.
- Designed the user interface using Bootstrap library and React API.
- Implemented the database via MongoDB and hosted in Azure.
- Built the frontend and backend connection with MVC Pattern.
- Developed the project structure with Decorator Pattern and implemented algorithms with Strategy Pattern in JavaScript.
- Provide status updates on where the request is in the overall process and any communication with fiscal staff.
- Designed different dashboards for different access levels (submitters, approvers, fiscal staff, admin).
- Implemented various functionalities for corresponding users such as communication channels, request approval, and workflow management.
- Provide status updates on where the request is in the overall process and any communication with fiscal staff.
- Designed the user interface using Bootstrap library and React API.
- Implemented the database via MongoDB and hosted in Azure.
- Built the frontend and backend connection with MVC Pattern.
- Implemented functionality for communication between the requester and approver.
- Allow communication between the staff and the requester.
- Provide status updates on where the request is in the overall process and any communication with fiscal staff.

BACK-END

- API is written using NODE JS and Express libraries.
- API can accept following HTTP requests:
  - GET
  - POST
  - PUT
  - DELETE
- NPM is used to automatically install dependencies and run backend script on deployment to server using Github pipeline.
- Highly Scalable due to MVC Architecture.
- Allows to store any types of form, even new forms without any modification to the backend program or database structure.
- Low Maintenance.
- Allows hierarchy of user access to the system.
- Ability to localize budgets to a specific subunit in the system.
- Ability to handle large amount of data (> 400 TB).
- Without any performance hit.
- Faster backend response time (Averaging ~150ms).
- This web portal consists of 4 main levels:
  - Requester layer to submit requests
  - Approver layer to approve requests under a subunit
  - Fiscal staff layer to deal with requests under a unit
  - Administrator layer to handle all higher levels settings
- Built a highly secure backend service with Azure, and control the response time to less than 200 ms.
- Through the user-friendly interface, easier and clearer processes can be provided for all users in all aspects.

ACKNOWLEDGEMENT

We would like to thank:
- Our mentors Ted Hanson and Bridge O Faherty for their support and suggestions during the long and strenuous journey of completing the project.
- All participating members who hard-work has made this project a possibility.
- Shruti Misra and Payman Arabshahi for their endless guidance and help.
- All participating members who hard-work has made this project a possibility.
- Shruti Misra and Payman Arabshahi for their endless guidance and help.