APIs that need to be done:

The yellow ones are already implemented.

Based on the Prisma schema, we would need the following API endpoints:

- Test

* GET test/ - creates some sample data including the roles

- User

* GET /user: returns a list of all users
* POST /user: creates a new user
* GET /user/{userID}: returns a specific user by userID
* GET user/[id] - returns a user based on query parameter of [id]
* POST/user/[id]: returns the users in a project based on query param of projectID
* POST/user/get/fullName: returns a user based on body params: {firstName, lastName}
* PUT /user/{userID}: updates a specific user by userID
* DELETE /user/{userID}: deletes a specific user by userID

- Project

* GET /projects: returns a list of all projects
* POST /projects: creates a new project
* POST/project/get: returns all the projects associated with a user
* Body params: {netID}
* POST/project/[projectNum]: returns a project based on query parameter of [projectNum]
* GET/project/{projectID}: returns a specific project by projectID
* PUT /projects/{projectID}: updates a specific project by projectID
* DELETE /projects/{projectID}: deactivates a specific project by projectID

- Request

* GET/request-form/get:: returns a list of all requests
* POST /request-form :: creates a new request
* POST request-form/get - returns the requests in a project that is associated to a user
  + body params: {netID}
* POST request-form/update – updates the fields of a request
  + body params: {requestID, itemID, description, url, partNumber, quantity, unitPrice}
  + optional params: {orderID, orderNumber, trackingInfo, shippingCost, processID, status, vendorID}
* GET /requests/{requestID}: returns a specific request by requestID
* PUT /requests/{requestID}: updates a specific request by requestID
* DELETE /requests/{requestID}: deletes a specific request by requestID

- RequestItem

* GET /requestItems: returns a list of all request items.
* POST /requestItems: creates a new request item
* GET /requestItems/{itemID}: returns a specific request item by itemID
* PUT /requestItems/{itemID}: updates a specific request item by itemID
* DELETE /requestItems/{itemID}: deletes a specific request item by itemID

- Order

* GET /orders: returns a list of all orders
* POST /orders: creates a new order
* GET /orders/{orderID}: returns a specific order by orderID
* PUT /orders/{orderID}: updates a specific order by orderID
* DELETE /orders/{orderID}: deletes a specific order by orderID

- OtherExpenses

* GET /otherExpenses: returns a list of all other expenses
* POST /otherExpenses: creates a new other expenses
* GET /otherExpenses/{expenseID}: returns a specific other expense by expenseID
* PUT /otherExpenses/{expenseID}: updates a specific other expense by expenseID
* DELETE /otherExpenses/{expenseID}: deletes a specific other expense by expenseID

- RequestUpload

* GET /requestUploads: returns a list of all request uploads
* POST /requestUploads: creates a new request upload
* GET /requestUploads/{uploadID}: returns a specific request upload by uploadID
* PUT /requestUploads/{uploadID}: updates a specific request upload by uploadID
* DELETE /requestUploads/{uploadID}: deletes a specific request upload by uploadID

-Reimbursement

* GET /reimbursements - Get a list of all reimbursements
* GET /reimbursements/{reimbursementID} - Get a specific reimbursement by ID
* POST /reimbursement-form/get - returns the reimbursements in a project associated to a user
  + body params: {netID}
* POST /reimbursement-form/- Create a new reimbursement
* POST reimbursement-form/update – updates the fields of a request
  + body params: {requestID, itemID, description, url, partNumber, quantity, unitPrice}
* PUT /reimbursements/{reimbursementID} - Update a specific reimbursement by ID
* DELETE /reimbursements/{reimbursementID} - Deactivate a specific reimbursement by ID

-ReimbursementItem

* GET /reimbursementItems- Get a list of all reimbursement items
* GET /reimbursementItems/{itemID} - Get a specific reimbursement item by ID
* POST /reimbursementItems- Create a new reimbursement item
* PUT /reimbursementItems/{itemID} - Update a specific reimbursement item by ID
* DELETE /reimbursementItems/{itemID} - Delete a specific reimbursement item by ID

-ReimbursementUpload

* GET /reimbursementUploads- Get a list of all reimbursement uploads
* GET /reimbursementUploads/{uploadID} - Get a specific reimbursement upload by ID
* POST /reimbursementUploads- Create a new reimbursement upload
* PUT /reimbursementUploads/{uploadID} - Update a specific reimbursement upload by ID
* DELETE /reimbursementUploads/{uploadID} Delete a specific reimbursement upload by ID

-Process

* GET /processes- Get a list of all processes
* GET /processes/{processID}- Get a specific process by ID
* POST /processes- Create a new process (NOT NEEDED)
* POST/process/update – updates the process of a request/reimbursement
  + Body params: {netID, requestID, comment, status}
* PUT /processes/{processID}- Update a specific process by ID
* DELETE /processes/{processID}- Delete a specific process by ID

-Vendor

* GET /vendor - creates the list of vendors
* GET /vendor/get - Get a list of all vendors
* GET /vendors/{vendorID} - Get a specific vendor by ID
* POST /vendors - Create a new vendor
* PUT /vendors/{vendorID} -Update a specific vendor by ID
* DELETE /vendors/{vendorID} - Delete a specific vendor by ID

-Role:

* POST /roles: Create a new role.
* GET /roles: Get a list of all roles.
* GET /roles/{roleId}: Get a single role by ID.
* PUT /roles/{roleId}: Update an existing role by ID.
* DELETE /roles/{roleId}: Delete a role by ID.

-previousRecords:

* POST /previousRecords: Create a new previous record.
* GET /previousRecords: Get a list of all previous records.
* GET /previousRecords/{recordId}: Get a single previous record by ID.
* PUT /previousRecords/{recordId}: Update an existing previous record by ID.
* DELETE /previousRecords/{recordId}: Delete a previous record by ID.

-departments:

* POST /departments: Create a new department.
* GET /departments: Get a list of all departments.
* GET /departments/{departmentId}: Get a single department by ID.
* PUT /departments/{departmentId}: Update an existing department by ID.
* DELETE /departments/{departmentId}: Delete a department by ID.

-departmentUsers:

* POST /departmentUsers: Create a new department user.
* GET /departmentUsers: Get a list of all department users.
* GET /departmentUsers/{departmentId}/{userId}: Get a single department user by department ID and user ID.
* PUT /departmentUsers/{departmentId}/{userId}: Update an existing department user by department ID and user ID.
* DELETE /departmentUsers/{departmentId}/{userId}: Delete a department user by department ID and user ID.

-worksOn:

* POST /worksOn: Create a new works on entry to add a user to a project.
  + Body params: {netID, projectNum}
* POST/worksOn/deactivate/: Deactivate user in a project by updating endDate.
  + Body params: {netID, projectNum, startDate, comments (optional)}
* GET /worksOn/currentProjects/{userId}: Get the current projects of a user.
* GET /worksOn/currentUsers/{projectId}: Get the current users in a project.
* GET /worksOn/pastProjects/{userId}: Get the past projects of a user.
* GET /worksOn/pastUsers/{projectId}: Get the past users in a project.

-projectDepartments

* POST /projectDepartments: Create a new project department.
* GET /projectDepartments: Get a list of all project departments.
* GET /projectDepartments/{departmentId}/{projectId}: Get a single project department by department ID and project ID.
* PUT /projectDepartments/{departmentId}/{projectId}: Update an existing project department by department ID and project ID.
* DELETE /projectDepartments/{departmentId}/{projectId}: Delete a project department by department ID and project ID.