Creating a banking system with Meteor.js involves building a full-stack application that includes both frontend and backend components. Below is a basic outline for implementing the described functionality.

### Stage 1: Technical Assignment

#### Prerequisites:

- Install Meteor.js: [Installation Guide](https://www.meteor.com/install)

- Set up a GitHub account for version control: [GitHub](https://github.com/)

#### Project Structure:

```

/banking-system

├── /client

│ └── /templates

│ ├── admin.html

│ ├── borrower.html

│ ├── lender.html

│ └── user.html

├── /server

│ └── main.js

├── /lib

│ └── collections.js

├── /public

│ └── /css

│ └── main.css

├── /tests

└── package.json

```

#### Instructions:

1. \*\*User Registration and Roles:\*\*

- Implement user registration using email. You can use the Meteor `accounts-base` package for user authentication.

2. \*\*Roles:\*\*

- Allow users to choose roles during registration (admin, borrower, lender).

3. \*\*Loan Request and Past Loans:\*\*

- For Borrowers:

- Implement the ability for borrowers to request a loan.

- Display existing past loans on the borrower's dashboard.

4. \*\*Loan Confirmation and Past Payments:\*\*

- For Lenders:

- Allow lenders to confirm and pay loans.

- Display existing past payments on the lender's dashboard.

5. \*\*Dashboard Updates:\*\*

- Ensure that the change of status (loan request, loan confirmation, payments) is reflected on each user's dashboard.

6. \*\*Admin View:\*\*

- Implement an admin view to see complete transactions.

7. \*\*Deployment and GitHub:\*\*

- Deploy the application (e.g., using Meteor Galaxy, Heroku, or another hosting service).

- Share the deployed application for functional testing.

- Share the code through GitHub for quality check.

### API Endpoints Test Cases:

#### Users:

1. \*\*POST /api/register:\*\*

- Register a new user.

- Include parameters: email, password, role.

2. \*\*GET /api/users/:userId:\*\*

- Retrieve user details by userId.

#### Loans:

3. \*\*POST /api/loans/request:\*\*

- Borrower requests a loan.

- Include parameters: userId, amount, description.

4. \*\*GET /api/loans/borrower/:userId:\*\*

- Retrieve past loans of a borrower.

5. \*\*POST /api/loans/confirm:\*\*

- Lender confirms and pays a loan.

- Include parameters: loanId, userId.

6. \*\*GET /api/loans/lender/:userId:\*\*

- Retrieve past payments of a lender.

#### Admin:

7. \*\*GET /api/admin/transactions:\*\*

- Retrieve all transactions for admin view.

Remember to handle authentication and authorization properly. Use Meteor methods for secure server-side operations. Perform testing at each stage to ensure functionality.

### Submission:

- Deploy the application for functional testing.

- Share the GitHub repository link.

- Include test cases for the provided API endpoints.

- Prepare a brief documentation explaining the application structure and key decisions made during development.