CHI ALPHA: CHRISTIAN FELLOWSHIP RESOURCE APPLICATION

Team: Web Ready

Team Members: Vincent Yen, Justin Lee

COMPLETED WORK

Database:

The announcement and events database functions should be operational, but still require integration testing.

Pages:

The announcement and events pages are working, but need tweaking so that it can be integrated with database querying

WORK IN PROGRESS

Database:

Currently working on creating the server that will be the middle-man which will communicate between mobile app and database. Figuring out how to upload media files to the database so that it can be integrated with the media and about us pages.

Testing:

Creating test cases to test each feature that we've created so far.

Integration:

Combining the different working parts so that they work together such as the database and the mobile app.
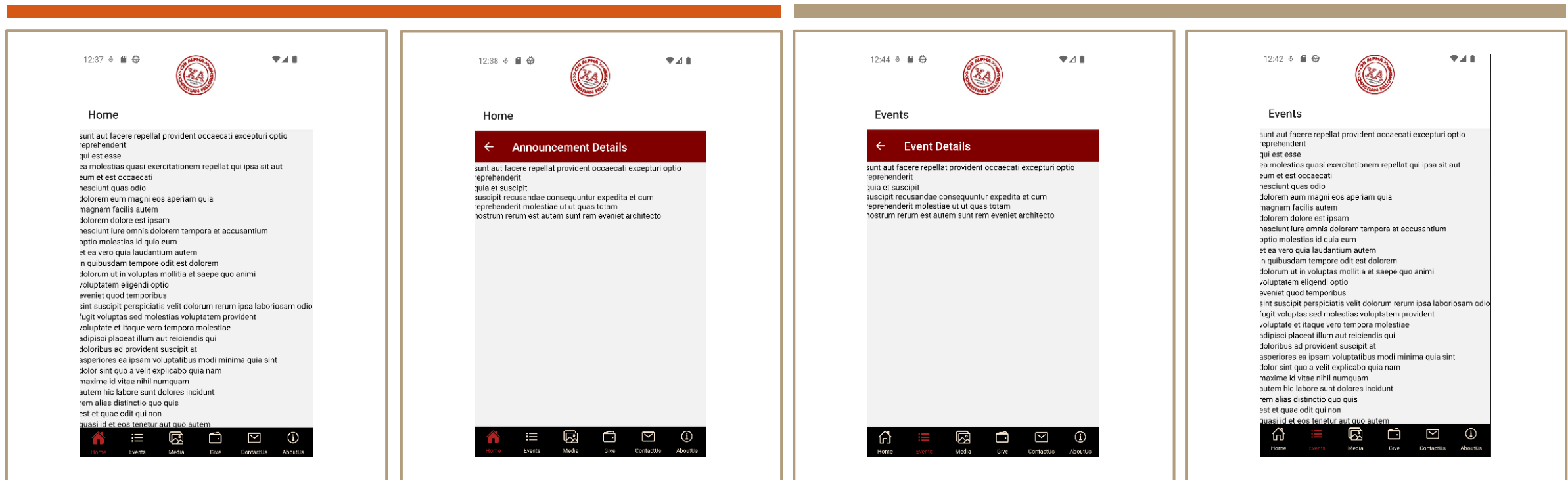
FUTURE WORK

There are a few things we expect to work on once the current work in progress is completed, they are as follows:

- **Admin pages**: These pages will allow the content creator to modify and add things to the database so that they can be displayed to the general user later. These things include being able to add/edit announcements and events among other things. These pages will only be available for viewing by the content creator, general users will not be able to access or see these pages.

- **Giving page**: Integrating the page with Square app so that it can serve as a payment service for donations.

- **Testing on IOS**: We've primarily been working with an Android emulator so that we could at least build this for one type of device, but we plan on testing it on an IOS emulator at some point so that all mobile devices will be covered.

- **Chat Application**: The chat application may be the last feature we will work on because we wanted to be able to deliver as many features as we can in our limited time. The chat application would take a large amount of time to be able to test and deliver.

DATABASE MODEL

Schema	Entries
User	{username, password, access level}
Announcement	{subject, description}
Event	{title, description, list of questions}
Question	{question, list of responses}
Response	{name, answer}
Media Directory	{name, list of media}
Media	{media file}
About	{name, description, picture}

USER INTERFACE SAMPLES

Server API

Route Functions

Home Function
Aspect to be tested: app.get("/api/")
Expected Result: • Should call function main.index(req, res)   • Should return a JSON response.   Observed Result: • index function is called. Response was returned from function.   Last Tested: 12/2/2022   Test Case Requirement: NodeJS, MongooseJS, BcryptJS, MongoDB

User Functions
Aspect to be tested: app.post("/api/user/login")
Expected Result: • Should call function main.loginUser(req, res)   • Should return a JSON response.   Observed Result: • loginUser function is called. Response was returned from function.   Last Tested: 12/2/2022   Test Case Requirement: NodeJS, MongooseJS, BcryptJS, MongoDB
Aspect to be tested: app.post("/api/user/new")   Expected Result: • Should call function main.newUser(req, res)   • Should return a JSON response.   Observed Result: • newUser function is called. Response was returned from function.   Last Tested: 12/2/2022   Test Case Requirement: NodeJS, MongooseJS, BcryptJS, MongoDB

SAMPLE TEST CASES