**CS345 Project 2 Progress Report**

**1. App entry point (BroomiesApp.swift)**

@main

struct BroomiesApp: App {

var body: some Scene {

WindowGroup {

ContentView()

}

}

}

struct BroomiesApp\_Previews: PreviewProvider {

static var previews: some View {

ContentView()

.previewDevice("iPhone 14 Pro")

}

}

**2. Content & Landing Page**

// ContentView.swift

struct ContentView: View {

var body: some View {

NavigationStack {

LandingPageView()

}

}

}

// LandingPageView.swift

struct LandingPageView: View {

var body: some View {

VStack {

Spacer()

Image("broomies\_icon")

.resizable()

.scaledToFit()

.frame(width: 200)

Text("BROOMIES")

.font(.largeTitle).bold()

.padding(.top, 8)

Spacer()

NavigationLink("Get Started") {

HouseNameView()

}

.buttonStyle(.borderedProminent)

.padding(.bottom, 40)

}

.background(AppColors.tan)

}

}

**3. House-name input (HouseNameView.swift)**

struct HouseNameView: View {

@State private var houseName = ""

var body: some View {

VStack(spacing: 30) {

Image("house\_icon")

.resizable()

.scaledToFit()

.frame(width: 120)

.padding()

.background(AppColors.blue)

.cornerRadius(20)

Text("WHAT’s YOUR HOUSE NAME?")

.font(.title2).bold()

.multilineTextAlignment(.center)

TextField("Latinx House", text: $houseName)

.textFieldStyle(.roundedBorder)

.padding(.horizontal, 40)

Spacer()

NavigationLink("Next") {

RoomiesView(houseName: houseName)

}

.disabled(houseName.isEmpty)

.buttonStyle(.borderedProminent)

}

.padding(.top, 60)

.background(AppColors.tan)

}

}

**4. Roommate-entry UI (RoomiesView.swift)**

struct RoomiesView: View {

var houseName: String

@State private var roommates: [String] = ["", ""]

var body: some View {

VStack(spacing: 16) {

Text("WHO ARE THE BROOMIES?")

.font(.title2).bold()

ForEach(roommates.indices, id: \.self) { i in

HStack {

Image("roommate\_icon")

.resizable()

.frame(width: 40, height: 40)

TextField("Name", text: Binding(

get: { roommates[i] },

set: { roommates[i] = $0 }

))

.textFieldStyle(.roundedBorder)

Button {

roommates.remove(at: i)

} label: {

Image(systemName: "xmark.circle.fill")

}

}

.padding(.horizontal, 24)

}

Button {

roommates.append("")

} label: {

Label("Add Roomie", systemImage: "plus")

}

.padding(.top, 8)

Spacer()

NavigationLink("Next") {

ChoresView(roommates: roommates.filter { !$0.isEmpty }.shuffled())

}

.disabled(roommates.allSatisfy { $0.isEmpty })

.buttonStyle(.borderedProminent)

}

.padding(.top, 40)

.background(AppColors.tan)

}

}

**5. Chore model & selection (ChoresView.swift)**

struct Chore: Identifiable, Hashable {

let id = UUID()

let name: String

let icon: String

}

let allChores: [Chore] = [

.init(name: "TRASH", icon: "trash\_icon"),

.init(name: "KITCHEN", icon: "kitchen\_icon"),

.init(name: "FRIDGE", icon: "fridge\_icon"),

.init(name: "STAIRS", icon: "stairs\_icon"),

.init(name: "LIVING ROOM", icon: "livingroom\_icon"),

.init(name: "BATHROOM", icon: "bathroom\_icon"),

]

struct ChoresView: View {

let roommates: [String]

@State private var selected: Set<Chore> = []

private let columns = [GridItem(.adaptive(minimum: 120), spacing: 20)]

var body: some View {

VStack {

Text("YOUR CHORES")

.font(.title).bold()

Text("SELECT ALL THAT APPLY.")

.font(.caption)

.foregroundStyle(.secondary)

.padding(.bottom, 16)

LazyVGrid(columns: columns, spacing: 20) {

ForEach(allChores) { chore in

Button {

if selected.contains(chore) {

selected.remove(chore)

} else {

selected.insert(chore)

}

} label: {

VStack {

Image(chore.icon)

.resizable()

.scaledToFit()

.frame(width: 80, height: 80)

.padding(12)

.background(selected.contains(chore) ? .green.opacity(0.2) : AppColors.tan)

.cornerRadius(15)

Text(chore.name)

.font(.footnote).bold()

}

}

}

}

.padding(.horizontal, 24)

Spacer()

NavigationLink("Assign Tasks") {

TasksView(roommates: roommates, chores: Array(selected))

}

.disabled(selected.isEmpty)

.buttonStyle(.borderedProminent)

.padding(.bottom, 32)

}

.padding(.top, 40)

.background(AppColors.tan)

}

}

**6. Task assignment list (TasksView.swift)**

struct TasksView: View {

let roommates: [String]

let chores: [Chore]

var body: some View {

VStack(spacing: 12) {

Text("TASK ASSIGNMENTS")

.font(.title2).bold()

if roommates.isEmpty || chores.isEmpty {

Text("No roommates or chores selected.")

} else {

List {

ForEach(chores.indices, id: \.self) { i in

HStack {

Image(chores[i].icon)

.resizable()

.frame(width: 30, height: 30)

Text(chores[i].name)

Spacer()

Text(roommates[i % roommates.count])

.fontWeight(.semibold)

}

}

}

.listStyle(.plain)

}

}

.background(AppColors.tan)

}

}

**7. Color palette (AppColors.swift)**

enum AppColors {

static let tan = Color(red: 1.0, green: 0.96, blue: 0.90)

static let blue = Color(red: 0.20, green: 0.30, blue: 0.40)

}