✅ First: What is an MVP for IKEA Kreativ?

MVP = Minimum Viable Product that allows a user to:

1. Upload a room photo
2. Clean the background (remove furniture)
3. Place 3D furniture into the scene
4. Drag/rotate furniture in the photo
5. See the **total cost** of placed furniture
6. Enjoy a minimal but smooth UX

These are your major functional areas:

1. **Room Background Setup**
2. **Furniture Placement**
3. **UI Tools & Interaction**
4. **Image Upload & Cleanup**
5. **Environment Setup & Infra**

🖼️ 1. Room Background Setup

User Story	Acceptance Criteria
As a user, I want to upload a photo of my room so I can visualize furniture inside it.	Image is uploaded successfully; a preview is shown.
As a user, I want the background of the room to be cleaned automatically (remove objects)	API returns a clean image; result is displayed accurately. - Calls a background removal API (e.g., remove.bg or LaMa). - API returns a cleaned image. - Cleaned image is applied to the 3D environment as a backdrop.
As a user, I want to adjust/crop the uploaded photo to fit the 3D scene properly.	User can crop/scale; updated image syncs with canvas. - Cropping tool available with zoom and drag.

User Story	Acceptance Criteria
	- Image crop is saved and synced to canvas.
As a user, I want to upload a photo of my room so I can visualize furniture in my space.	- Upload button available on landing page or UI modal. - Only image file types accepted (JPG, PNG). - Uploaded image appears on canvas.

🪑 2. Furniture Placement

User Story	Acceptance Criteria
As a user, I want to browse furniture items and preview them in my space.	A list of models loads; each can be selected and viewed. - A catalog/grid of .glb models is shown. - On selection, model appears in 3D scene.
As a user, I want to drag, rotate, and position furniture freely inside the room.	Models respond to mouse/touch events accurately. - Furniture can be moved using mouse/touch. - Rotate handles or hotkeys allow rotation. - Position/rotation updates are real-time.
As a user, I want to remove furniture if I change my mind.	Clicking delete removes the model from the scene. - Each item has a delete or "X" control. - Removing an item updates the total price.
As a user, I want to see the price of each furniture item I add.	- Each item has a price tag displayed on hover or in a sidebar. - Total cost updates dynamically as items are added/removed.

🛠️ 3. UI Tools & Interaction

User Story	Acceptance Criteria
As a user, I want to see helpful tips or a tutorial for how to use the tool.	Tutorial modal or tooltips are shown on first use.
As a user, I want undo/redo buttons so I can fix mistakes.	Undo reverts last move; redo re-applies it.
As a user, I want to reset the scene to start from scratch.	A reset button clears all placed models and reloads canvas. - "Reset Scene" button clears all placed items. - Background image and canvas are cleared.
As a user, I want to see a list of all added furniture in one place.	- Sidebar or modal lists all items. - Includes name, quantity, price per item, total per item. - Grand total is displayed.
As a user, I want to add furniture items to a cart or see where to buy them.	- Each furniture item has a "View Product" or "Buy" button. - Button links to external product page or mock store. - Optional: Items added to a shopping cart UI with quantities and checkout link.

💠 4. Image Upload & Cleanup (API)

User Story	Acceptance Criteria
As a user, I want the uploaded image to be cleaned by AI (remove.bg, etc.)	Cleaned image loads with removed background.
As a user, I want the app to alert me if the image is too low quality or corrupted.	Error handling shows helpful message; no crash occurs.
As a user, I want to upload a room photo using my phone camera, so I can work directly from my space.	- Mobile users can launch the device camera from within the app (via HTML5 input or native bridge). - Captured image is previewed and processed like a normal upload.

User Story	Acceptance Criteria
	- Fallback to manual upload if camera not available.

⚙️ 5. Environment Setup & Infra (Dev only)

Dev Story	Acceptance Criteria
As a dev, I want to configure environment variables for API keys securely.	.env.local file works; sensitive data is not exposed.
As a dev, I want to load .glb files from public/assets or external URLs.	Model URLs are dynamic and easily configurable.
As a dev, I want to version control the base setup with clear commits.	Git tracks base setup cleanly; no sensitive files committed.