

Step-2: Brainstorm, Idea Listing and Grouping

2 Brainstorm

- Use **VGG16** for accurate image classification using transfer learning.
- Add a **confidence score** with the prediction result.
- Label predictions as **"Good to Eat"** or **"Don't Eat."**
- Design a simple **Flask web interface** for uploading images.
- Store **user feedback** to improve the model over time.
- Support **28 distinct classes** for fruits and vegetables.
- Handle low-resolution or blurry image uploads with error messages.
- Keep the model **lightweight** for faster prediction on web.
- Add a **future React frontend** for a responsive user experience.
- Explore **model compression** for mobile-friendly versions.

20 minutes

Person 1 → Mariya Suguna

- Use labeled subfolders (healthy/unhealthy) for each fruit/vegetable class.
- Ensure balanced train and validation data.
- Augment images to improve generalization.

Person 3 → M.M Bhavesh H.R.K

- Build upload and result pages with Flask and Jinja2.
- Display confidence score and "Good to Eat" message.
- Handle image upload and file validation.

Person 2 → Durga Chaitra

- Use VGG16 for transfer learning with fine-tuning.
- Evaluate using accuracy, precision, recall, F1 score.
- Save best-performing model checkpoints.

Person 4 → Poornima D.C

- Create a feedback form to collect user reviews.
- Store feedback in a JSON file for later use.
- Test for error cases like blurry images or wrong file formats.

3 Group ideas

Model & Accuracy

- VGG16 for feature extraction
- Confidence score display
- Support for 28-class categorization
- Model compression for deployment

User Interface

- Flask backend with HTML templates
- Future upgrade to React frontend
- Image upload with validation and result page

User Feedback & Improvement

- Feedback collection via form
- Store feedback in JSON or database
- Retrain model periodically based on feedback

Error Handling

- Handle low-quality image uploads
- File type validation (only image formats allowed)

20 minutes

TIP You can select a sticky note and hit the pencil icon to start drawing

TIP Add customizable tags to sticky notes to make it easier to find, remove, organize, and categorize important ideas as themes within your mind.

Step-3: Idea Prioritization

4 Prioritize

- **High Importance + High Feasibility (Top-Right Corner)**
 - Use VGG16 for image classification via transfer learning
 - Upload images through Flask interface
 - Display prediction, confidence score, and recommendation
 - Collect user feedback through feedback form
 - Support all 28 fruit and vegetable categories
- **High Importance + Low Feasibility (Top-Left Corner)**
 - Real-time camera input for instant classification
 - Retrain the model periodically based on user feedback
 - Compress model for mobile or edge device compatibility
- **Low Importance + High Feasibility (Bottom-Right Corner)**
 - Add visual enhancements to static HTML pages
 - Store feedback in JSON instead of setting up a database
 - Display extra prediction details (e.g., time taken, model version)
- **Low Importance + Low Feasibility (Bottom-Left Corner)**
 - Switch to a different CNN model like ResNet during current phase
 - Add multi-language UI support in the MVP version

20 minutes

TIP Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H** key on the keyboard.

Importance

If each of these tasks could get done without any difficulty or cost,

Feasibility

Regardless of their importance (Cost, time, effort, complexity, etc.)