# Setting up tasks and resources

**1. Identifying my resources**

- **Time:** How much time can you realistically dedicate to this project per week?
- **Skills:** What are your strengths (e.g., Python, Flask/Django, cryptography)? Where do you need to learn or improve?
- **Tools:** What software, libraries, or services will you use? (e.g., Python, Flask/Django, SQLite/MySQL, cryptography library - specify which one, IDE, Git, project management software).
- **Budget:** Although this is a personal project, consider any costs like hosting, domain name (if you plan to deploy it), or paid software/services.

**2. Breaking down the project into tasks**

- List all the tasks required
    - Set up Flask/Django project.
    - Design database schema.
    - Implement user authentication (registration, login, logout).
    - Implement password hashing with bcrypt/Argon2.
    - Implement AES encryption for messages.
    - Implement RSA for key exchange.
    - Create inbox/outbox functionality.
    - Develop web interface (HTML, CSS, JavaScript).
    - Write unit tests.
    - Write documentation.
    - Deploy the application.

**4. Assign Resources to Tasks**

- Allocate time for 5-7 PM 2 hours of developing.
- I have all tools for my web app
- May need to pay for hosting

**5. Prioritize Tasks**

- Identify critical tasks that need to be completed first (e.g., setting up the basic framework, implementing user authentication).
- Use a project management technique like MoSCoW (Must have, Should have, Could have, Won't have) to prioritize features.

**6. Track Progress and Adjust**

- Regularly monitor your progress and compare it to your estimates.

- If you're falling behind, identify the reasons why and adjust your plan accordingly. This might involve re-prioritizing tasks, allocating more time, or seeking help.
- Use a project management tool (e.g., Trello, Asana, Jira, Clockify) to track tasks, deadlines, and progress.

**Example Resource Allocation Table**

| Task | Resource: Time (hours) | Resource: Skills | Resource: Tools |
|---|---|---|---|
| Set up Flask project | 2 | Python, Flask | VS Code, Git |
| Implement user registration | 8 | Python, Flask, Databases | VS Code, Git, SQLite |
| Implement bcrypt password hashing | 4 | Python, bcrypt library | VS Code, Git |
| Implement AES encryption | 6 | Python, AES library | VS Code, Git |
| Create inbox/outbox functionality | 10 | Python, Flask, Databases, | VS Code, Git, SQLite, HTML/CSS |
| Write basic documentation | 4 | Writing, Markdown | Text editor, Git |